Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'ctermid.3p' command

$ man ctermid.3p

CTERMID(3P) POSIX Programmer's Manual CTERMID(3P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

ctermid ? generate a pathname for the controlling terminal

SYNOPSIS

```
#include <stdio.h>
char *ctermid(char *s);
```

DESCRIPTION

The ctermid() function shall generate a string that, when used as a pathname, refers to the current controlling terminal for the current process. If ctermid() returns a pathname, access to the file is not guaranteed.

The ctermid() function need not be thread-safe if called with a NULL parameter.

RETURN VALUE

If s is a null pointer, the string shall be generated in an area that may be static, the address of which shall be returned. The application shall not modify the string returned. The returned pointer might be in? validated or the string content might be overwritten by a subsequent

call to ctermid(). The returned pointer might also be invalidated if the calling thread is terminated. If s is not a null pointer, s is as? sumed to point to a character array of at least L_ctermid bytes; the string is placed in this array and the value of s shall be returned.

The symbolic constant L_ctermid is defined in <stdio.h>, and shall have a value greater than 0.

The ctermid() function shall return an empty string if the pathname that would refer to the controlling terminal cannot be determined, or if the function is unsuccessful.

ERRORS

No errors are defined.

The following sections are informative.

EXAMPLES

Determining the Controlling Terminal for the Current Process

The following example returns a pointer to a string that identifies the controlling terminal for the current process. The pathname for the ter? minal is stored in the array pointed to by the ptr argument, which has a size of L_ctermid bytes, as indicated by the term argument.

```
#include <stdio.h>
...
char term[L_ctermid];
char *ptr;
ptr = ctermid(term);
```

APPLICATION USAGE

The difference between ctermid() and ttyname() is that ttyname() must be handed a file descriptor and return a path of the terminal associ? ated with that file descriptor, while ctermid() returns a string (such as "/dev/tty") that refers to the current controlling terminal if used as a pathname.

RATIONALE

L_ctermid must be defined appropriately for a given implementation and must be greater than zero so that array declarations using it are ac? cepted by the compiler. The value includes the terminating null byte.

Conforming applications that use multiple threads cannot call ctermid() with NULL as the parameter. If s is not NULL, the ctermid() function generates a string that, when used as a pathname, refers to the current controlling terminal for the current process. If s is NULL, the return value of ctermid() is undefined.

There is no additional burden on the programmer?changing to use a hypothetical thread-safe version of ctermid() along with allocating a buffer is more of a burden than merely allocating a buffer. Application code should not assume that the returned string is short, as some implementations have more than two pathname components before reaching a logical device name.

FUTURE DIRECTIONS

None.

SEE ALSO

ttyname()

The Base Definitions volume of POSIX.1?2017, <stdio.h>

COPYRIGHT