NAME

is_term_resized, **resize_term**, **resizeterm -** change the curses terminal size

SYNOPSIS

#include <curses.h>

bool is_term_resized(int *lines*, **int** *columns***);**
int resize_term(int *lines*, **int** *columns***);**
int resizeterm(int *lines*, **int** *columns***);**

DESCRIPTION

This is an extension to the curses library. It provides callers with a hook into the **ncurses** data to resize windows, primarily for use by programs running in an X Window terminal (e.g., xterm).

resizeterm

The function **resizeterm** resizes the standard and current windows to the specified dimensions, and adjusts other bookkeeping data used by the **ncurses** library that record the window dimensions such as the **LINES** and **COLS** variables.

resize_term

Most of the work is done by the inner function **resize_term**. The outer function **resizeterm** adds bookkeeping for the **SIGWINCH** handler, as well as repainting the soft-key area (see **slk_touch**(3X)).

When resizing the windows, **resize_term** blank-fills the areas that are extended. The calling application should fill in these areas with appropriate data.

The **resize_term** function attempts to resize all windows. However, due to the calling convention of pads, it is not possible to resize these without additional interaction with the application.

When resizing windows, **resize_term** recursively adjusts subwindows, keeping them within the updated parent window's limits. If a top-level window happens to extend to the screen's limits, then on resizing the window, **resize_term** will keep the window extending to the corresponding limit, regardless of whether the screen has shrunk or grown.

is_term_resized

A support function **is_term_resized** is provided so that applications can check if the **resize_term** function would modify the window structures. It returns **TRUE** if the windows would be modified, and **FALSE** otherwise.

RETURN VALUE

Except as noted, these functions return the integer **ERR** upon failure and **OK** on success. They will fail if either of the dimensions are less than or equal to zero, or if an error occurs while (re)allocating memory for the windows.

NOTES

While these functions are intended to be used to support a signal handler (i.e., for **SIGWINCH**), care should be taken to avoid invoking them in a context where **malloc** or **realloc** may have been interrupted, since it uses those functions.

If ncurses is configured to supply its own **SIGWINCH** handler,

- on receipt of a **SIGWINCH**, the handler sets a flag

- which is tested in **wgetch**(3X) and **doupdate**,

- in turn, calling the **resizeterm** function,

- which **ungetch**'s a **KEY_RESIZE** which will be read on the next call to **wgetch**.

  The **KEY_RESIZE** alerts an application that the screen size has changed, and that it should repaint special features such as pads that cannot be done automatically.

  Calling **resizeterm** or **resize_term** directly from a signal handler is unsafe. This indirect method is used to provide a safe way to resize the ncurses data structures.

If the environment variables **LINES** or **COLUMNS** are set, this overrides the library's use of the window size obtained from the operating system. Thus, even if a **SIGWINCH** is received, no screen size change may be recorded.

PORTABILITY

It is possible to resize the screen with SVr4 curses, by

- exiting curses with **endwin**(3X) and

- resuming using **refresh**(3X).

Doing that clears the screen and is visually distracting.

This extension of ncurses was introduced in mid-1995. It was adopted in NetBSD curses (2001) and PDCurses (2003).

SEE ALSO

curs_getch(3X), **curs_variables**(3X), **wresize**(3X).

AUTHOR

Thomas Dickey (from an equivalent function written in 1988 for BSD curses).