NAME

curs_slk, slk_init, slk_set, slk_refresh, slk_noutrefresh, slk_label, slk_clear, slk_restore, slk_touch, slk_attron, slk_attrset, slk_attroff – curses soft label routines

SYNOPSIS

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

#include <curses.h>

int slk_init(int *fmt***);**

int slk_set(int *labnum***, char** **label***, int** *fmt***);**

int slk_refresh(void);

int slk_noutrefresh(void);

char *slk_label(int *labnum***);**

int slk_clear(void);

int slk_restore(void);

int slk_touch(void);

int slk_attron(chtype *attrs***);**

int slk_attrset(chtype *attrs***);**

int slk_attroff(chtype *attrs***);**

MT-LEVEL

Unsafe

DESCRIPTION

curses manipulates the set of soft function-key labels that exist on many terminals. For those terminals that do not have soft labels, **curses** takes over the bottom line of **stdscr**, reducing the size of **stdscr** and the variable **LINES**. **curses** standardizes on eight labels of up to eight characters each.

To use soft labels, the **slk_init( )** routine must be called before **initscr( )** or **newterm( )** is called. If **initscr( )** eventually uses a line from **stdscr** to emulate the soft labels, then *fmt* determines how the labels are arranged on the screen. Setting *fmt* to **0** indicates a 3-2-3 arrangement of the labels; **1** indicates a 4-4 arrangement.

With the **slk_set( )** routine, *labnum* is the label number, from **1** to **8**. *label* is the string to be put on the label, up to eight characters in length. A null string or a null pointer sets up a blank label. *fmt* is either **0**, **1**, or **2**, indicating whether the label is to be left-justified, centered, or right-justified, respectively, within the label.

The **slk_refresh( )** and **slk_noutrefresh( )** routines correspond to the **wrefresh( )** and **wnoutrefresh( )** routines.

With the **slk_label( )** routine, the current label for label number *labnum* is returned with leading and trailing blanks stripped.

With the **slk_clear( )** routine, the soft labels are cleared from the screen.

With the **slk_restore( )** routine, the soft labels are restored to the screen after a **slk_clear( )** is performed.

With the **slk_touch( )** routine, all the soft labels are forced to be output the next time a **slk_noutrefresh( )** is performed.

The **slk_attron( )**, **slk_attrset( )**, and **slk_attroff( )** routines correspond to **attron( )**, **attrset( )**, and **attroff( )**. They have an effect only if soft labels are simulated on the bottom line of the screen.

RETURN VALUES

Routines that return an integer return **ERR** upon failure and an integer value other than **ERR** upon successful completion.

slk_label( ) returns **NULL** on error.

SEE ALSO

curs_attr(3X), **curs_initscr**(3X), **curs_refresh**(3X), **curses**(3X)

NOTES

The header **<curses.h>** automatically includes the headers **<stdio.h>** and **<unctrl.h>**.

Most applications would use **slk_noutrefresh( )** because a **wrefresh( )** is likely to follow soon.