NAME

curs_kernel, def_prog_mode, def_shell_mode, reset_prog_mode, reset_shell_mode, resetty, savetty, getsyx, setsyx, ripoffline, curs_set, napms – low-level curses routines

SYNOPSIS

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

#include <curses.h>

int def_prog_mode(void);

int def_shell_mode(void);

int reset_prog_mode(void);

int reset_shell_mode(void);

int resetty(void);

int savetty(void);

int getsyx(int *y*, **int** *x***);**

int setsyx(int *y*, **int** *x***);**

int ripoffline(int *line*, **int (*init)(WINDOW *, int));**

int curs_set(int *visibility***);**

int napms(int *ms***);**

MT-LEVEL

Unsafe

DESCRIPTION

The following routines give low-level access to various **curses** functionality. Theses routines typically are used inside library routines.

The **def_prog_mode( )** and **def_shell_mode( )** routines save the current terminal modes as the ''program'' (in **curses**) or ''shell'' (not in **curses)** state for use by the **reset_prog_mode( )** and **reset_shell_mode( )** routines. This is done automatically by **initscr( )**.

The **reset_prog_mode( )** and **reset_shell_mode( )** routines restore the terminal to ''program'' (in **curses**) or ''shell'' (out of **curses**) state. These are done automatically by **endwin( )** and, after an **endwin( )**, by **doupdate( )**, so they normally are not called.

The **resetty( )** and **savetty( )** routines save and restore the state of the terminal modes. **savetty( )** saves the current state in a buffer and **resetty( )** restores the state to what it was at the last call to **savetty( )**.

With the **getsyx( )** routine, the current coordinates of the virtual screen cursor are returned in *y* and *x*. If **leaveok( )** is currently **TRUE**, then –**1**,–**1** is returned. If lines have been removed from the top of the screen, using **ripoffline( )**, *y* and *x* include these lines; therefore, *y* and *x* should be used only as arguments for **setsyx( )**.

With the **setsyx( )** routine, the virtual screen cursor is set to *y*, *x*. If *y* and *x* are both –**1**, then **leaveok( )** is set. The two routines **getsyx( )** and **setsyx( )** are designed to be used by a library routine, which manipulates **curses** windows but does not want to change the current position of the program's cursor. The library routine would call **getsyx( )** at the beginning, do its manipulation of its own windows, do a **wnoutrefresh( )** on its windows, call **setsyx( )**, and then call **doupdate( )**.

The **ripoffline( )** routine provides access to the same facility that **slk_init( )** (see **curs_slk**(3X)) uses to reduce the size of the screen. **ripoffline( )** must be called before **initscr( )** or **newterm( )** is called. If *line* is positive, a line is removed from the top of **stdscr( )**; if *line* is negative, a line is removed from the bottom. When this is done inside **initscr( )**, the routine **init( )** (supplied by the user) is called with two arguments: a window pointer to the one-line window that has been allocated and an integer with the number of columns in the window. Inside this initialization routine, the integer variables **LINES** and **COLS** (defined in **<curses.h>**) are not guaranteed to be accurate and **wrefresh( )** or **doupdate( )** must not be called. It is allowable to call **wnoutrefresh( )** during the initialization routine.

ripoffline( ) can be called up to five times before calling **initscr( )** or **newterm( )**.

With the **curs_set( )** routine, the cursor state is set to invisible, normal, or very visible for *visibility* equal to **0**, **1**, or **2** respectively. If the terminal supports the *visibility* requested, the previous *cursor* state is returned; otherwise, **ERR** is returned.

The **napms( )** routine is used to sleep for *ms* milliseconds.

RETURN VALUES

Except for **curs_set( )**, these routines always return **OK**. **curs_set( )** returns the previous cursor state, or **ERR** if the requested *visibility* is not supported.

SEE ALSO

curs_initscr(3X), **curs_outopts**(3X), **curs_refresh**(3X), **curs_scr_dump**(3X), **curs_slk**(3X), **curses**(3X)

NOTES

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

Note that **getsyx( )** is a macro, so an ampersand (**&**) is not necessary before the variables *y* and *x*.