NAME

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

```
#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);

void getsyx(int y, int x);
void setsyx(int y, int x);

int ripoffline(int line, int (*init)(WINDOW *, int));
int curs_set(int visibility);
int napms(int ms);
```

DESCRIPTION

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

def_prog_mode, def_shell_mode

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**. There is one such save area for each screen context allocated by **newterm**.

reset_prog_mode, reset_shell_mode

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**(3X) and, after an **endwin**, by **doupdate**, so they normally are not called.

resetty, savetty

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**.

getsyx

The **getsyx** routine returns the current coordinates of the *virtual screen* cursor 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**.

Few applications will use this feature, most use **getyx** instead.

setsyx

The **setsyx** routine sets the *virtual screen* cursor 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**.

Few applications will use this feature, most use **wmove** instead.

ripoffline

ripoffline provides access to the same facility that **slk_init**(3X) uses to reduce the size of the screen. **ripoffline** must be called before **initscr** or **newterm** is called, to prepare these initial actions:

- 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 the resulting initialization 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**.

curs_set

The **curs_set** routine sets the cursor state 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.

napms

napms sleeps for *ms* milliseconds. If *ms* exceeds 30,000 (thirty seconds), it is capped at that value.

RETURN VALUE

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.

X/Open defines no error conditions. In this implementation

def_prog_mode, **def_shell_mode**, **reset_prog_mode**, **reset_shell_mode**

return an error if the terminal was not initialized, or if the I/O call to obtain the terminal settings fails.

ripoffline

returns an error if the maximum number of ripped-off lines exceeds the maximum (5).

NOTES

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

Older SVr4 man pages warn that the return value of **curs_set** "is currently incorrect". This implementation gets it right, but it may be unwise to count on the correctness of the return value anywhere else.

Both *ncurses* and SVr4 will call **curs_set** in **endwin** if **curs_set** has been called to make the cursor other than normal, i.e., either invisible or very visible. There is no way for *ncurses* to determine the initial cursor state to restore that.

PORTABILITY

The *virtual screen* functions **setsyx** and **getsyx** are not described in X/Open Curses, Issue 4. All other functions are as described in X/Open Curses.

The SVr4 documentation describes **setsyx** and **getsyx** as having return type int. This is misleading, as they are macros with no documented semantics for the return value.

If interrupted, *ncurses* restarts **napms**. That, and the limitation to 30 seconds, are different from other

implementations.

SEE ALSO

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