NAME

curs_initscr, initscr, newterm, endwin, isendwin, set_term, delscreen – curses screen initialization and manipulation routines

SYNOPSIS

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

#include <curses.h>

WINDOW *initscr(void);

int endwin(void);

int isendwin(void);

SCREEN *newterm(char **type*, **FILE** **outfd*, **FILE** **infd*);

SCREEN *set_term(SCREEN **new*);

void delscreen(SCREEN* *sp*);

MT-LEVEL

Unsafe

DESCRIPTION

initscr( ) is almost always the first routine that should be called (the exceptions are **slk_init( )**, **filter( )**, **ripoffline( )**, **use_env( )** and, for multiple-terminal applications, **newterm( )**.) This determines the terminal type and initializes all **curses** data structures. **initscr( )** also causes the first call to **refresh( )** to clear the screen. If errors occur, **initscr( )** writes an appropriate error message to standard error and exits; otherwise, a pointer is returned to **stdscr( )**. If the program needs an indication of error conditions, **newterm()** should be used instead of **initscr( )**; **initscr( )** should only be called once per application.

A program that outputs to more than one terminal should use the **newterm( )** routine for each terminal instead of **initscr( )**. A program that needs an indication of error conditions, so it can continue to run in a line-oriented mode if the terminal cannot support a screen-oriented program, would also use this routine. The routine **newterm( )** should be called once for each terminal. It returns a variable of type **SCREEN** * which should be saved as a reference to that terminal. The arguments are the *type* of the terminal to be used in place of **$TERM**, a file pointer for output to the terminal, and another file pointer for input from the terminal (if *type* is **NULL**, **$TERM** will be used). The program must also call **endwin( )** for each terminal being used before exiting from curses. If **newterm( )** is called more than once for the same terminal, the first terminal referred to must be the last one for which **endwin( )** is called.

A program should always call **endwin( )** before exiting or escaping from **curses** mode temporarily. This routine restores tty modes, moves the cursor to the lower left-hand corner of the screen and resets the terminal into the proper non-visual mode. Calling **refresh( )** or **doupdate( )** after a temporary escape causes the program to resume visual mode.

The **isendwin( )** routine returns **TRUE** if **endwin( )** has been called without any subsequent calls to **wrefresh( )**, and **FALSE** otherwise.

The **set_term( )** routine is used to switch between different terminals. The screen reference **new** becomes the new current terminal. The previous terminal is returned by the routine. This is the only routine which manipulates **SCREEN** pointers; all other routines affect only the current terminal.

The **delscreen( )** routine frees storage associated with the **SCREEN** data structure. The **endwin( )** routine does not do this, so **delscreen( )** should be called after **endwin( )** if a particular **SCREEN** is no longer needed.

RETURN VALUES

endwin( ) returns the integer **ERR** upon failure and **OK** upon successful completion.

Routines that return pointers always return **NULL** on error.

SEE ALSO

curs_kernel(3X), **curs_refresh**(3X), **curs_slk**(3X), **curs_util**(3X), **curses**(3X)

NOTES

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

Note that **initscr( )** and **newterm( )** may be macros.