NAME

curs_color, start_color, init_pair, init_color, has_colors, can_change_color, color_content, pair_content – curses color manipulation routines

SYNOPSIS

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

#include <curses.h>

int start_color(void);

int init_pair(short *pair***, short** *fg***, short** *bg***);**

int init_color(short *color***, short** *red***, short** *green***, short** *blue***);**

bool has_colors(void);

bool can_change_color(void);

int color_content(short *color***, short** **redp***, short** **greenp***, short** **bluep***);**

int pair_content(short *pair***, short** **fgp***, short** **bgp***);**

MT-LEVEL

Unsafe

DESCRIPTION

Overview

curses provides routines that manipulate color on color alphanumeric terminals. To use these routines **start_color( )** must be called, usually right after **initscr( )**. See **curs_initscr**(3X). Colors are always used in pairs (referred to as color-pairs). A color-pair consists of a foreground color (for characters) and a background color (for the field on which the characters are displayed). A programmer initializes a color-pair with the routine **init_pair**. After it has been initialized, **COLOR_PAIR(***n***)**, a macro defined in **<curses.h>**, can be used in the same ways other video attributes can be used. If a terminal is capable of redefining colors, the programmer can use the routine **init_color( )** to change the definition of a color. The routines **has_colors( )** and **can_change_color( )** return **TRUE** or **FALSE**, depending on whether the terminal has color capabilities and whether the programmer can change the colors. The routine **color_content( )** allows a programmer to identify the amounts of red, green, and blue components in an initialized color. The routine **pair_content( )** allows a programmer to find out how a given color-pair is currently defined.

Routine Descriptions

The **start_color( )** routine requires no arguments. It must be called if the programmer wants to use colors, and before any other color manipulation routine is called. It is good practice to call this routine right after **initscr( )**. **start_color( )** initializes eight basic colors (black, red, green, yellow, blue, magenta, cyan, and white), and two global variables, **COLORS** and **COLOR_PAIRS** (respectively defining the maximum number of colors and color-pairs the terminal can support). It also restores the colors on the terminal to the values they had when the terminal was just turned on.

The **init_pair( )** routine changes the definition of a color-pair. It takes three arguments: the number of the color-pair to be changed, the foreground color number, and the background color number. The value of the first argument must be between **1** and **COLOR_PAIRS**–1. The value of the second and third arguments must be between 0 and

COLORS. If the color-pair was previously initialized, the screen is refreshed and all occurrences of that color-pair is changed to the new definition.

The **init_color( )** routine changes the definition of a color. It takes four arguments: the number of the color to be changed followed by three RGB values (for the amounts of red, green, and blue components). The value of the first argument must be between **0** and **COLORS**. (See the section **Colors** for the default color index.) Each of the last three arguments must be a value between 0 and 1000. When **init_color( )** is used, all occurrences of that color on the screen immediately change to the new definition.

The **has_colors( )** routine requires no arguments. It returns **TRUE** if the terminal can manipulate colors; otherwise, it returns **FALSE**. This routine facilitates writing terminal-independent programs. For example, a programmer can use it to decide whether to use color or some other video attribute.

The **can_change_color( )** routine requires no arguments. It returns **TRUE** if the terminal supports colors and can change their definitions; other, it returns **FALSE**. This routine facilitates writing terminal-independent programs.

The **color_content( )** routine gives users a way to find the intensity of the red, green, and blue (RGB) components in a color. It requires four arguments: the color number, and three addresses of **short**s for storing the information about the amounts of red, green, and blue components in the given color. The value of the first argument must be between 0 and **COLORS**. The values that are stored at the addresses pointed to by the last three arguments are between 0 (no component) and 1000 (maximum amount of component).

The **pair_content( )** routine allows users to find out what colors a given color-pair consists of. It requires three arguments: the color-pair number, and two addresses of **short**s for storing the foreground and the background color numbers. The value of the first argument must be between 1 and **COLOR_PAIRS**–1. The values that are stored at the addresses pointed to by the second and third arguments are between 0 and **COLORS**.

Colors

In **<curses.h>** the following macros are defined. These are the default colors. **curses** also assumes that **COLOR_BLACK** is the default background color for all terminals.

COLOR_BLACK
COLOR_RED
COLOR_GREEN
COLOR_YELLOW
COLOR_BLUE
COLOR_MAGENTA
COLOR_CYAN
COLOR_WHITE

RETURN VALUES

All routines that return an integer return **ERR** upon failure and **OK** upon successful completion.

SEE ALSO

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

NOTES

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