NAME

curses – CRT screen handling and optimization package

SYNOPSIS

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

#include <curses.h>

MT-LEVEL

Unsafe

DESCRIPTION

The **curses** library routines give the user a terminal-independent method of updating character screens with reasonable optimization.

The **curses** package allows: overall screen, window and pad manipulation; output to windows and pads; reading terminal input; control over terminal and **curses** input and output options; environment query routines; color manipulation; use of soft label keys; terminfo access; and access to low-level **curses** routines.

To initialize the routines, the routine **initscr( )** or **newterm( )** must be called before any of the other routines that deal with windows and screens are used. The routine **endwin( )** must be called before exiting. To get character-at-a-time input without echoing (most interactive, screen oriented programs want this), the following sequence should be used:

initscr,cbreak,noecho;

Most programs would additionally use the sequence:

nonl,intrflush(stdscr,FALSE),keypad(stdscr,TRUE);

Before a **curses** program is run, the tab stops of the terminal should be set and its initialization strings, if defined, must be output. This can be done by executing the **tput init** command after the shell environment variable **TERM** has been exported. (See **terminfo**(4) for further details.)

The **curses** library permits manipulation of data structures, called *windows*, which can be thought of as two-dimensional arrays of characters representing all or part of a CRT screen. A default window called **stdscr**, which is the size of the terminal screen, is supplied. Others may be created with **newwin**(3X).

Windows are referred to by variables declared as **WINDOW** *. These data structures are manipulated with routines described on 3X pages (whose names begin "curs_"). Among which the most basic routines are **move**(3X) and **addch**(3X). More general versions of these routines are included with names beginning with **w**, allowing the user to specify a window. The routines not beginning with **w** affect **stdscr**.

After using routines to manipulate a window, **refresh**(3X) is called, telling **curses** to make the user's CRT screen look like **stdscr**. The characters in a window are actually of type **chtype**, (character and attribute data) so that other information about the character may also be stored with each character.

Special windows called *pads* may also be manipulated. These are windows which are not constrained to the size of the screen and whose contents need not be completely displayed. See **curs_pad**(3X) for more information.

In addition to drawing characters on the screen, video attributes and colors may be included, causing the characters to show up in such modes as underlined, in reverse video, or in color on terminals that support such display enhancements. Line drawing characters may be specified to be output. On input, **curses** is also able to translate arrow and function keys that transmit escape sequences into single values. The video attributes, line drawing characters, and input values use names, defined in **<curses.h>**, such as **A_REVERSE**, **ACS_HLINE**, and **KEY_LEFT**.

If the environment variables **LINES** and **COLUMNS** are set, or if the program is executing in a window environment, line and column information in the environment will override information read by *terminfo*. This would effect a program running in an AT&T 630 layer, for example, where the size of a screen is changeable.

If the environment variable **TERMINFO** is defined, any program using **curses** checks for a local terminal definition before checking in the standard place. For example, if **TERM** is set to **att4424**, then the compiled terminal definition is found in

/usr/share/lib/terminfo/a/att4424.

(The '**a**' is copied from the first letter of **att4424** to avoid creation of huge directories.) However, if **TERMINFO** is set to **$HOME/myterms**, **curses** first checks

$HOME/myterms/a/att4424,

and if that fails, it then checks

/usr/share/lib/terminfo/a/att4424.

This is useful for developing experimental definitions or when write permission in **/usr/share/lib/terminfo** is not available.

The integer variables **LINES** and **COLS** are defined in **<curses.h>** and will be filled in by **initscr** with the size of the screen. The constants **TRUE** and **FALSE** have the values **1** and **0**, respectively.

The **curses** routines also define the **WINDOW** ∗ variable **curscr** which is used for certain low-level operations like clearing and redrawing a screen containing garbage. The **curscr** can be used in only a few routines.

International Functions

The number of bytes and the number of columns to hold a character from the supplementary character set is locale-specific (locale category **LC_CTYPE**) and can be specified in the character class table.

For editing, operating at the character level is entirely appropriate. For screen formatting, arbitrary movement of characters on screen is not desirable.

Overwriting characters (**addch**, for example) operates on a screen level. Overwriting a character by a character that requires a different number of columns may produce *orphaned columns*. These orphaned columns are filled with background characters.

Inserting characters (**insch**, for example) operates on a character level (that is, at the character boundaries). The specified character is inserted right before the character, regardless of which column of a character the cursor points to. Before insertion, the cursor position is adjusted to the first column of the character.

As with inserting characters, deleting characters (**delch**, for example) operates on a character level (that is, at the character boundaries). The character at the cursor is deleted whichever column of the character the cursor points to. Before deletion, the cursor position is adjusted to the first column of the character.

A *multi-column* character cannot be put on the last column of a line. When such attempts are made, the last column is set to the background character. In addition, when such an operation creates orphaned columns, the orphaned columns are filled with background characters.

Overlapping and overwriting a window follows the operation of overwriting characters around its edge. The orphaned columns, if any, are handled as in the character operations.

The cursor is allowed to be placed anywhere in a window. If the insertion or deletion is made when the cursor points to the second or later column position of a character that holds multiple columns, the cursor is adjusted to the first column of the character before the insertion or deletion.

Routine and Argument Names

Many **curses** routines have two or more versions. The routines prefixed with **w** require a window argument. The routines prefixed with **p** require a pad argument. Those without a prefix generally use **stdscr**.

The routines prefixed with **mv** require an *x* and *y* coordinate to move to before performing the appropriate action. The **mv** routines imply a call to **move**(3X) before the call to the other routine. The coordinate *y* always refers to the row (of the window), and *x* always refers to the column. The upper left-hand corner is always (0,0), not (1,1).

The routines prefixed with **mvw** take both a window argument and *x* and *y* coordinates. The window argument is always specified before the coordinates.

In each case, *win* is the window affected, and *pad* is the pad affected; *win* and *pad* are always pointers to type **WINDOW**

Option setting routines require a Boolean flag *bf* with the value **TRUE** or **FALSE**; *bf* is always of type **bool**. The variables *ch* and *attrs* below are always of type **chtype**. The types **WINDOW**, **SCREEN**, **bool**, and **chtype** are defined in **<curses.h>**. The type **TERMINAL** is defined in **<term.h>**. All other arguments are integers.

Routine Name Index

The following table lists each **curses** routine and the name of the manual page on which it is described.

curses Routine Name	Manual Page Name
addch	curs_addch(3X)
addchnstr	curs_addchstr(3X)
addchstr	curs_addchstr(3X)
addnstr	curs_addstr(3X)
addnwstr	curs_addwstr(3X)
addstr	curs_addstr(3X)
addwch	curs_addwch(3X)
addwchnstr	curs_addwchstr(3X)
addwchstr	curs_addwchstr(3X)
addwstr	curs_addwstr(3X)
adjcurspos	curs_alecompat(3X)
attroff	curs_attr(3X)
attron	curs_attr(3X)
attrset	curs_attr(3X)
baudrate	curs_termattrs(3X)
beep	curs_beep(3X)
bkgd	curs_bkgd(3X)
bkgdset	curs_bkgd(3X)
border	curs_border(3X)
box	curs_border(3X)
can_change_color	curs_color(3X)
cbreak	curs_inopts(3X)
clear	curs_clear(3X)
clearok	curs_outopts(3X)
clrtobot	curs_clear(3X)
clrtoeol	curs_clear(3X)
color_content	curs_color(3X)
copywin	curs_overlay(3X)
curs_set	curs_kernel(3X)
def_prog_mode	curs_kernel(3X)
def_shell_mode	curs_kernel(3X)
del_curterm	curs_terminfo(3X)
delay_output	curs_util(3X)
delch	curs_delch(3X)
deleteln	curs_deleteln(3X)
delscreen	curs_initscr(3X)
delwin	curs_window(3X)
derwin	curs_window(3X)
doupdate	curs_refresh(3X)
dupwin	curs_window(3X)

echo	**curs_inopts(3X)**
echochar	**curs_addch(3X)**
echowchar	**curs_addwch(3X)**
endwin	**curs_initscr(3X)**
erase	**curs_clear(3X)**
erasechar	**curs_termattrs(3X)**
filter	**curs_util(3X)**
flash	**curs_beep(3X)**
flushinp	**curs_util(3X)**
getbegyx	**curs_getyx(3X)**
getch	**curs_getch(3X)**
getmaxyx	**curs_getyx(3X)**
getnwstr	**curs_getwstr(3X)**
getparyx	**curs_getyx(3X)**
getstr	**curs_getstr(3X)**
getsyx	**curs_kernel(3X)**
getwch	**curs_getwch(3X)**
getwin	**curs_util(3X)**
getwstr	**curs_getwstr(3X)**
getyx	**curs_getyx(3X)**
halfdelay	**curs_inopts(3X)**
has_colors	**curs_color(3X)**
has_ic	**curs_termattrs(3X)**
has_il	**curs_termattrs(3X)**
idcok	**curs_outopts(3X)**
idlok	**curs_outopts(3X)**
immedok	**curs_outopts(3X)**
inch	**curs_inch(3X)**
inchnstr	**curs_inchstr(3X)**
inchstr	**curs_inchstr(3X)**
init_color	**curs_color(3X)**
init_pair	**curs_color(3X)**
initscr	**curs_initscr(3X)**
innstr	**curs_instr(3X)**
innwstr	**curs_inwstr(3X)**
insch	**curs_insch(3X)**
insdelln	**curs_deleteln(3X)**
insertln	**curs_deleteln(3X)**
insnstr	**curs_insstr(3X)**
insnwstr	**curs_inswstr(3X)**
insstr	**curs_insstr(3X)**
instr	**curs_instr(3X)**
inswch	**curs_inswch(3X)**
inswstr	**curs_inswstr(3X)**
intrflush	**curs_inopts(3X)**

inwch	**curs_inwch(3X)**
inwchnstr	**curs_inwchstr(3X)**
inwchstr	**curs_inwchstr(3X)**
inwstr	**curs_inwstr(3X)**
is_linetouched	**curs_touch(3X)**
is_wintouched	**curs_touch(3X)**
isendwin	**curs_initscr(3X)**
keyname	**curs_util(3X)**
keypad	**curs_inopts(3X)**
killchar	**curs_termattrs(3X)**
leaveok	**curs_outopts(3X)**
longname	**curs_termattrs(3X)**
meta	**curs_inopts(3X)**
move	**curs_move(3X)**
movenextch	**curs_alecompat(3X)**
moveprevch	**curs_alecompat(3X)**
mvaddch	**curs_addch(3X)**
mvaddchnstr	**curs_addchstr(3X)**
mvaddchstr	**curs_addchstr(3X)**
mvaddnstr	**curs_addstr(3X)**
mvaddnwstr	**curs_addwstr(3X)**
mvaddstr	**curs_addstr(3X)**
mvaddwch	**curs_addwch(3X)**
mvaddwchnstr	**curs_addwchstr(3X)**
mvaddwchstr	**curs_addwchstr(3X)**
mvaddwstr	**curs_addwstr(3X)**
mvcur	**curs_terminfo(3X)**
mvdelch	**curs_delch(3X)**
mvderwin	**curs_window(3X)**
mvgetch	**curs_getch(3X)**
mvgetnwstr	**curs_getwstr(3X)**
mvgetstr	**curs_getstr(3X)**
mvgetwch	**curs_getwch(3X)**
mvgetwstr	**curs_getwstr(3X)**
mvinch	**curs_inch(3X)**
mvinchnstr	**curs_inchstr(3X)**
mvinchstr	**curs_inchstr(3X)**
mvinnstr	**curs_instr(3X)**
mvinnwstr	**curs_inwstr(3X)**
mvinsch	**curs_insch(3X)**
mvinsnstr	**curs_insstr(3X)**
mvinsnwstr	**curs_inswstr(3X)**
mvinsstr	**curs_insstr(3X)**
mvinstr	**curs_instr(3X)**
mvinswch	**curs_inswch(3X)**

mvinswstr	**curs_inswstr(3X)**
mvinwch	**curs_inwch(3X)**
mvinwchnstr	**curs_inwchstr(3X)**
mvinwchstr	**curs_inwchstr(3X)**
mvinwstr	**curs_inwstr(3X)**
mvprintw	**curs_printw(3X)**
mvscanw	**curs_scanw(3X)**
mvwaddch	**curs_addch(3X)**
mvwaddchnstr	**curs_addchstr(3X)**
mvwaddchstr	**curs_addchstr(3X)**
mvwaddnstr	**curs_addstr(3X)**
mvwaddnwstr	**curs_addwstr(3X)**
mvwaddstr	**curs_addstr(3X)**
mvwaddwch	**curs_addwch(3X)**
mvwaddwchnstr	**curs_addwchstr(3X)**
mvwaddwchstr	**curs_addwchstr(3X)**
mvwaddwstr	**curs_addwstr(3X)**
mvwdelch	**curs_delch(3X)**
mvwgetch	**curs_getch(3X)**
mvwgetnwstr	**curs_getwstr(3X)**
mvwgetstr	**curs_getstr(3X)**
mvwgetwch	**curs_getwch(3X)**
mvwgetwstr	**curs_getwstr(3X)**
mvwin	**curs_window(3X)**
mvwinch	**curs_inch(3X)**
mvwinchnstr	**curs_inchstr(3X)**
mvwinchstr	**curs_inchstr(3X)**
mvwinnstr	**curs_instr(3X)**
mvwinnwstr	**curs_inwstr(3X)**
mvwinsch	**curs_insch(3X)**
mvwinsnstr	**curs_insstr(3X)**
mvwinsstr	**curs_insstr(3X)**
mvwinstr	**curs_instr(3X)**
mvwinswch	**curs_inswch(3X)**
mvwinswstr	**curs_inswstr(3X)**
mvwinwch	**curs_inwch(3X)**
mvwinwchnstr	**curs_inwchstr(3X)**
mvwinwchstr	**curs_inwchstr(3X)**
mvwinwstr	**curs_inwstr(3X)**
mvwprintw	**curs_printw(3X)**
mvwscanw	**curs_scanw(3X)**
napms	**curs_kernel(3X)**
newpad	**curs_pad(3X)**
newterm	**curs_initscr(3X)**
newwin	**curs_window(3X)**

nl	**curs_outopts(3X)**
nocbreak	**curs_inopts(3X)**
nodelay	**curs_inopts(3X)**
noecho	**curs_inopts(3X)**
nonl	**curs_outopts(3X)**
noqiflush	**curs_inopts(3X)**
noraw	**curs_inopts(3X)**
notimeout	**curs_inopts(3X)**
overlay	**curs_overlay(3X)**
overwrite	**curs_overlay(3X)**
pair_content	**curs_color(3X)**
pechochar	**curs_pad(3X)**
pechowchar	**curs_pad(3X)**
pnoutrefresh	**curs_pad(3X)**
prefresh	**curs_pad(3X)**
printw	**curs_printw(3X)**
putp	**curs_terminfo(3X)**
putwin	**curs_util(3X)**
qiflush	**curs_inopts(3X)**
raw	**curs_inopts(3X)**
redrawwin	**curs_refresh(3X)**
refresh	**curs_refresh(3X)**
reset_prog_mode	**curs_kernel(3X)**
reset_shell_mode	**curs_kernel(3X)**
resetty	**curs_kernel(3X)**
restartterm	**curs_terminfo(3X)**
ripoffline	**curs_kernel(3X)**
savetty	**curs_kernel(3X)**
scanw	**curs_scanw(3X)**
scr_dump	**curs_scr_dump(3X)**
scr_init	**curs_scr_dump(3X)**
scr_restore	**curs_scr_dump(3X)**
scr_set	**curs_scr_dump(3X)**
scroll	**curs_scroll(3X)**
scrollok	**curs_outopts(3X)**
set_curterm	**curs_terminfo(3X)**
set_term	**curs_initscr(3X)**
setscrreg	**curs_outopts(3X)**
setsyx	**curs_kernel(3X)**
setterm	**curs_terminfo(3X)**
setupterm	**curs_terminfo(3X)**
slk_attroff	**curs_slk(3X)**
slk_attron	**curs_slk(3X)**
slk_attrset	**curs_slk(3X)**
slk_clear	**curs_slk(3X)**

slk_init	**curs_slk(3X)**
slk_label	**curs_slk(3X)**
slk_noutrefresh	**curs_slk(3X)**
slk_refresh	**curs_slk(3X)**
slk_restore	**curs_slk(3X)**
slk_set	**curs_slk(3X)**
slk_touch	**curs_slk(3X)**
srcl	**curs_scroll(3X)**
standend	**curs_attr(3X)**
standout	**curs_attr(3X)**
start_color	**curs_color(3X)**
subpad	**curs_pad(3X)**
subwin	**curs_window(3X)**
syncok	**curs_window(3X)**
termattrs	**curs_termattrs(3X)**
termname	**curs_termattrs(3X)**
tgetent	**curs_termcap(3X)**
tgetflag	**curs_termcap(3X)**
tgetnum	**curs_termcap(3X)**
tgetstr	**curs_termcap(3X)**
tgoto	**curs_termcap(3X)**
tigetflag	**curs_terminfo(3X)**
tigetnum	**curs_terminfo(3X)**
tigetstr	**curs_terminfo(3X)**
timeout	**curs_inopts(3X)**
touchline	**curs_touch(3X)**
touchwin	**curs_touch(3X)**
tparm	**curs_terminfo(3X)**
tputs	**curs_terminfo(3X)**
typeahead	**curs_inopts(3X)**
unctrl	**curs_util(3X)**
ungetch	**curs_getch(3X)**
ungetwch	**curs_getwch(3X)**
untouchwin	**curs_touch(3X)**
use_env	**curs_util(3X)**
vidattr	**curs_terminfo(3X)**
vidputs	**curs_terminfo(3X)**
vwprintw	**curs_printw(3X)**
vwscanw	**curs_scanw(3X)**
waddch	**curs_addch(3X)**
waddchnstr	**curs_addchstr(3X)**
waddchstr	**curs_addchstr(3X)**
waddnstr	**curs_addstr(3X)**
waddnwstr	**curs_addwstr(3X)**
waddstr	**curs_addstr(3X)**

waddwch	**curs_addwch(3X)**
waddwchnstr	**curs_addwchstr(3X)**
waddwchstr	**curs_addwchstr(3X)**
waddwstr	**curs_addwstr(3X)**
wadjcurspos	**curs_alecompat(3X)**
wattroff	**curs_attr(3X)**
wattron	**curs_attr(3X)**
wattrset	**curs_attr(3X)**
wbkgd	**curs_bkgd(3X)**
wbkgdset	**curs_bkgd(3X)**
wborder	**curs_border(3X)**
wclear	**curs_clear(3X)**
wclrtobot	**curs_clear(3X)**
wclrtoeol	**curs_clear(3X)**
wcursyncup	**curs_window(3X)**
wdelch	**curs_delch(3X)**
wdeleteln	**curs_deleteln(3X)**
wechochar	**curs_addch(3X)**
wechowchar	**curs_addwch(3X)**
werase	**curs_clear(3X)**
wgetch	**curs_getch(3X)**
wgetnstr	**curs_getstr(3X)**
wgetnwstr	**curs_getwstr(3X)**
wgetstr	**curs_getstr(3X)**
wgetwch	**curs_getwch(3X)**
wgetwstr	**curs_getwstr(3X)**
whline	**curs_border(3X)**
winch	**curs_inch(3X)**
winchnstr	**curs_inchstr(3X)**
winchstr	**curs_inchstr(3X)**
winnstr	**curs_instr(3X)**
winnwstr	**curs_inwstr(3X)**
winsch	**curs_insch(3X)**
winsdelln	**curs_deleteln(3X)**
winsertln	**curs_deleteln(3X)**
winsnstr	**curs_insstr(3X)**
winsnwstr	**curs_inswstr(3X)**
winsstr	**curs_insstr(3X)**
winstr	**curs_instr(3X)**
winswch	**curs_inswch(3X)**
winswstr	**curs_inswstr(3X)**
winwch	**curs_inwch(3X)**
winwchnstr	**curs_inwchstr(3X)**
winwchstr	**curs_inwchstr(3X)**
winwstr	**curs_inwstr(3X)**

wmove	**curs_move(3X)**
wmovenextch	**curs_alecompat(3X)**
wmoveprevch	**curs_alecompat(3X)**
wnoutrefresh	**curs_refresh(3X)**
wprintw	**curs_printw(3X)**
wredrawln	**curs_refresh(3X)**
wrefresh	**curs_refresh(3X)**
wscanw	**curs_scanw(3X)**
wscrl	**curs_scroll(3X)**
wsetscrreg	**curs_outopts(3X)**
wstandend	**curs_attr(3X)**
wstandout	**curs_attr(3X)**
wsyncdown	**curs_window(3X)**
wsyncup	**curs_window(3X)**
wtimeout	**curs_inopts(3X)**
wtouchln	**curs_touch(3X)**
wvline	**curs_border(3X)**

RETURN VALUES

Routines that return an integer return **ERR** upon failure and an integer value other than **ERR** upon successful completion, unless otherwise noted in the routine descriptions.

All macros return the value of the **w** version, except **setscrreg( )**, **wsetscrreg( )**, **getyx( )**, **getbegyx( )**, and **getmaxyx( )**. The return values of **setscrreg( )**, **wsetscrreg( )**, **getyx( ) ,** **getbegyx( )**, and **getmaxyx( )** are undefined (that is, these should not be used as the right-hand side of assignment statements).

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

SEE ALSO

terminfo(4) and 3X pages whose names begin with ''curs_'' for detailed routine descriptions.

NOTES

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