NAME

curs_util, unctrl, keyname, filter, use_env, putwin, getwin, delay_output, flushinp – curses miscellaneous utility routines

SYNOPSIS

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

#include <curses.h>

char *unctrl(chtype *c***);**

char *keyname(int *c***);**

int filter(void);

void use_env(char *bool***);**

int putwin(WINDOW **win*, **FILE** **filep***);**

WINDOW *getwin(FILE **filep***);**

int delay_output(int *ms***);**

int flushinp(void);

MT-LEVEL

Unsafe

DESCRIPTION

The **unctrl( )** macro expands to a character string which is a printable representation of the character *c*. Control characters are displayed in the ^X notation. Printing characters are displayed as is.

With the **keyname( )** routine, a character string corresponding to the key *c* is returned.

The **filter( )** routine, if used, is called before **initscr( )** or **newterm( )** are called. It makes **curses** think that there is a one-line screen. **curses** does not use any terminal capabilities that assume that they know on what line of the screen the cursor is positioned.

The **use_env( )** routine, if used, is called before **initscr( )** or **newterm( )** are called. When called with **FALSE** as an argument, the values of **lines** and **columns** specified in the *terminfo* database will be used, even if environment variables **LINES** and **COLUMNS** (used by default) are set, or if **curses** is running in a window (in which case default behavior would be to use the window size if **LINES** and **COLUMNS** are not set).

With the **putwin( )** routine, all data associated with window *win* is written into the file to which *filep* points. This information can be later retrieved using the **getwin( )** function.

The **getwin( )** routine reads window related data stored in the file by **putwin( )**. The routine then creates and initializes a new window using that data. It returns a pointer to the new window.

The **delay_output( )** routine inserts an *ms* millisecond pause in output. This routine should not be used extensively because padding characters are used rather than a CPU pause.

The **flushinp( )** routine throws away any typeahead that has been typed by the user and has not yet been read by the program.

RETURN VALUES

Except for **flushinp( )**, routines that return an integer return **ERR** upon failure and an integer value other than **ERR** upon successful completion.

flushinp( ) always returns **OK**.

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

SEE ALSO

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

NOTES

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

Note that **unctrl( )** is a macro, which is defined in **<unctrl.h>**.