NAME

newpad, **subpad**, **prefresh**, **pnoutrefresh**, **pechochar**, **pecho_wchar** − create and display *curses* pads

SYNOPSIS

```
#include <curses.h>

WINDOW *newpad(int nlines, int ncols);
WINDOW *subpad(WINDOW *parent, int nlines, int ncols,
    int begin_y, int begin_x);

int prefresh(WINDOW *pad, int pminrow, int pmincol,
    int sminrow, int smincol, int smaxrow, int smaxcol);
int pnoutrefresh(WINDOW *pad, int pminrow, int pmincol,
    int sminrow, int smincol, int smaxrow, int smaxcol);

int pechochar(WINDOW *pad, chtype ch);
int pecho_wchar(WINDOW *pad, const cchar_t *wch);
```

DESCRIPTION

A *curses pad* is like a window, except that it is not restricted by the screen size, and is not necessarily associated with a particular part of the screen. Pads can be used when a large window is needed, only part of which is to be visible on the screen. Pads are not automatically refreshed by scrolling or input-echoing operations.

Pads cannot be refreshed with **wrefresh**(3X); use **prefresh** or **pnoutrefresh** instead.

newpad

newpad creates and returns a pointer to a new pad data structure with the given number of lines, *nlines*, and columns, *ncols*.

subpad

subpad creates and returns a pointer to a subwindow within a pad with the given number of lines, *nlines*, and columns, *ncols*. Unlike **subwin**(3X), which uses screen coordinates, the new pad is placed at position (*begin_y*, *begin_x*) relative to its parent. Thus, changes made to one pad can affect both. When operating on a subpad, it is often necessary to call **touchwin**(3X) or **touchline**(3X) on *parent* before calling **prefresh**.

prefresh, pnoutrefresh

prefresh and **pnoutrefresh** are analogous to **wrefresh**(3X) and **wnoutrefresh**(3X) except that they operate on pads rather than windows. They require additional parameters are needed to indicate what portions of the pad and screen are involved.

- *pminrow* and *pmincol* specify the upper left-hand corner of a rectangular view of the pad.
- *sminrow*, *smincol*, *smaxrow*, and *smaxcol* specify the vertices of the rectangle to be displayed on the screen.

The lower right-hand corner of the rectangle to be displayed in the pad is calculated from the screen coordinates, since the rectangles must be the same size. Both rectangles must be entirely contained within their respective structures. *curses* treats negative values of any of these parameters as zero.

pechochar

pechochar is functionally equivalent to calling **waddch**(3X) followed by **prefresh**. It suggests to the *curses* optimizer that only a single character is being output; a considerable performance benefit may be thus enjoyed. The location of the character *ch* written to the pad is used to populate the arguments to **prefresh**.

pecho_wchar

pecho_wchar is functionally equivalent to calling **wadd_wch**(3X) followed by **prefresh**. It suggests to the *curses* optimizer that only a single wide character is being output; a considerable performance benefit may be thus enjoyed. The location of the character *wch* written to the pad is used to populate the arguments to **prefresh**.

RETURN VALUE

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

Functions that return pointers return a null pointer on failure, and set *errno* to **ENOMEM**.

In this implementation

prefresh and **pnoutrefresh**
: return **ERR** if the window pointer is null, or if the window is not really a pad or if the area to refresh extends off-screen or if the minimum coordinates are greater than the maximum.

pechochar
: returns **ERR** if the window is not really a pad, and the associated call to **wechochar** returns **ERR**.

pecho_wchar
: returns **ERR** if the window is not really a pad, and the associated call to **wecho_wchar** returns **ERR**.

NOTES

pechochar may be implemented as a macro.

curses documentation is traditionally averse to motivating the term "pad". The Apollo Aegis workstation operating system (*circa* 1981) supported a graphical pad feature.

- These graphical pads could be much larger than the computer's display.
- The read-only output from a command could be scrolled back to inspect and select text from the pad.

The two uses may be related.

PORTABILITY

X/Open Curses Issue 4 describes these functions. It specifies no error conditions for them.

SVr4 describes a successful return value only as "an integer value other than *ERR*".

The behavior of *subpad* if the parent window is not a pad is undocumented, and is not checked by the vendor Unix implementations.

- SVr4 *curses*'s *newpad* sets a flag in the *WINDOW* structure indicating that the window is a pad.

  However, it uses this information only in *waddch* (to decide if it should call *wrefresh*) and *wscrl* (to avoid scrolling a pad); its *wrefresh* does not check it to ensure that a pad is properly updated.

- Solaris *xcurses* checks whether a window is a pad in its *wnoutrefresh*, returning *ERR* in that case.

  However, it sets the flag on subwindows only if the parent window is a pad. Its *newpad* does not set this information. Consequently, the check never fails.

  It makes no comparable check in *pnoutrefresh* — though interestingly enough, a comment in the source code states that the lack of a check was an MKS extension.

- NetBSD 7 *curses* sets a flag in the *WINDOW* structure for *newpad* and *subpad*, aiding itself to distinguish between *wnoutrefresh* and *pnoutrefresh*.

  It does not check for the case where a subwindow is created in a pad using *subwin* or *derwin*.

  Its *dupwin* returns a regular window when duplicating a pad. Likewise, its *getwin* always returns a window, even if the saved data was from a pad.

ncurses:

- sets a flag in the *WINDOW* structure for *newpad* and *subpad*,
- allows a *subwin* or *derwin* call to succeed having a pad parent by forcing the subwindow to be a pad,
- checks in both *wnoutrefresh* and *pnoutrefresh* to ensure that pads and windows are handled distinctly, and

- ensures that *dupwin* and *getwin* treat pads versus windows consistently.

HISTORY

SVr2 (1984) introduced *newpad*, *prefresh*, and *pnoutrefresh*, documenting them in a single line each.

SVr3 (1987) added *subpad* and *pechochar*, and provided more extensive documentation.

The System V Interface Definition, Version 4 (1995), specified a function named *pechowchar*. This was a later addition to SVr4.*x*, not appearing in the first SVr4 (1989). It differs from X/Open's later *pecho_wchar* in that its *wstr* parameter was a *chtype* instead of a *wchar_t*, and was not *const*-qualified.

SEE ALSO

curses(3X), **curs_addch**(3X), **curs_refresh**(3X), **curs_touch**(3X)