

NCURSES 函数简要参考

BenBear

2004 年12 月18 日

目 录

§1 说明	1
§2 简介	1
§3 初始化与结束	2
§4 基本输入、输出函数预览	2
§5 函数命名、参数习惯	3
§6 getch()	4
§7 窗口WINDOW	4
§8 颜色	5
§9 Panel、Menus 和Forms	6
§10 其它	6

§1 说明

在这里写下对NCURSES 库使用的经验, 希望对你有用。由于条件的不足, 我只是在WMware Workstation 4.0.5 里安装的Debian Linux 3.0 r2 下使用过NCURSES, 所以这里所说的片面性在所难免, 还请细查。如有问题, 你可以给我发邮件: benbearchen@gmail.com。

§2 简介

NCURSES 是Linux 下进行终端界面开发的库, 其功能相当强大。它以WINDOW (窗口) 为基本要素, 在此基础上支持Panel、Menus 及Forms 等更加强大的元

素。更多的信息请参考资料。

§3 初始化与结束

当然了，为了使用NCURSES库，你应该#include <curses.h>，在编译时，应该加上-lncurses。

在使用NCURSES库前，必须先行初始化；用完后必须结束，否则终端可能会不正常。初始化是一系列的动作，内容有关功能键、输入回显、换行符、颜色等。

一般地，第一个函数调用必须是initscr()。这个函数初始化终端，然后清屏。它的参数为空，返回值一般也不用管。它会初始化一个叫stdscr的变量，stdscr有点像stdin或stdout的层次，它是NCURSES默认的窗口。

紧接着initscr()的应该是其它初始化，下面一一介绍。

即时按键。raw()和cbreak()两个函数都是用来关闭行缓冲以使按键可以马上被程序处理的。只是cbreak()可以在按Ctrl-C的时候退出程序，我一般用这个。两个函数都无参数。

使用功能键。keypad()可以用来打开一个窗口的功能键，从而使程序可以识别F1、F2、Up、Down等这样的按键。一般用keypad(stdscr, TRUE)来打开stdscr的这项功能。

按键回显。echo()和noecho()，后者可以关闭按键回显，这在使用getch()时特别有用。在程序中间，二者可以交替使用。两个函数都无参数。

换行符。nl()和nonl()，用来设置换行是对应一个'\n'还是'\r\n'。一般使用默认的要可以了，所以不需调用。两个函数都无参数。

启动颜色。has_colors()返回一个bool值，用来表示终端是否支持颜色；支持则返回真。如果支持颜色，那么就可以用start_color()来打开颜色功能。

综合这些，我的初始化代码一般是：

```
initscr ();
cbreak ();
noecho ();
start_color ();
keypad (stdscr, TRUE);
```

结束很简单，只需调用endwin();即可，但一定要记得调用。如下：

```
endwin();
```

§4 基本输入、输出函数预览

由于NCURSES已经接管了输入输出，所以<stdio.h>里标准I/O就不好再使用了；当然，你可以用，但很可能出错。

很高兴的是，NCURSES 中基本的输入输出设计得跟标准I/O 十分相似，比如输出有printw(), 输入有scanw(), 呵呵，只是把“f”换成了“w”，其它定义完全一样。

基本的输出函数有addch()、 addstr()、 printw(), 分别为输出一个字符、输出一个字符串和格式化输出。在这几个函数上会演化出很多函数。演化的两大要素是：坐标和窗口。

坐标是相对于窗口的左上角的，从(0, 0) 开始，但注意NCURSES 里的下标是(y, x) 表示的，这个务必记住。move(y, x); 可以直接移动stdscr 的光标到某一位置。窗口Window 是NCURSES 的基本操作对象，事实上所有的的输入输出都关联到窗口上，如果缺省的话，就是stdscr。

当然，从addstr() 到addnstr() 也是一种演化。

例如addch() 这一系列的函数有：addch()、 mvaddch()、 waddch() 和mvwaddch() 四个。其中“mv”表示move，也就是移动到某一坐标处再输出。而“w”表示Window，即向某一窗口输出。而move() 也可以变化成wmove()。

要注意的是，这些输出不是直接输出到屏幕上的，而是输出到某一窗口（当然缺省的是stdscr）。所以应该使用refresh() 刷新stdscr 的输出；对应的，用wrefresh() 可以刷新某一窗口的输出；wrefresh(stdscr) == refresh()。

基本的输入函数有getch()、 getstr()、 scanw(), 也可以演化出很多函数。

echo 的设置会对输入函数产生一定的影响，可能由于noecho 而使输入时没有回显。如果这样的话，可以先打开echo 然后再关闭。按照我看的资料，像scanw () 这样的函数，会自动打开echo 的，但实际上确实出现了没有回显的情况。

§5 函数命名、参数习惯

这些输入输出函数都是很规范的，下面总结一下。

如果没有前缀，那就表示在stdscr 的当前位置输入或输出。如果前缀是“w”，表示向某一个窗口的当前位置上操作。如果前缀是“mv”，表示要在stdscr 的某一位置上操作。如果前缀是“mvw”，表示在某一窗口的某一位置上进行操作。

参数的排列是这样的：([WINDOW,] [y, x,] arg1, agr2, ...)。也就是如果有窗口，那么窗口就是第一个参数；其次是如果有(y, x)，处于窗口后而在其它参数前。如addch() 系列是：

addch(ch)
mvaddch(y, x, ch)
waddch(win, ch)
mvwaddch (win, y, x, ch)

知道这些规则，对于记忆众多的函数大有益处。

§6 getch()

getch() 是极有用的函数，它可以读取功能键，所以几乎是必须用到的函数。如果echo()，那么getch() 是有回显的；否则，noecho() 下没有回显。只有在keypad(win, TRUE) 后，才能使用功能键。

getch() 无参数，返回值表示按键的值，可以是一个ASCII 值，也可能是一个功能键。功能键可以在man 3 getch 里查到，都是以KEY_ 开头的。下面列出常用的一些：

回退键	KEY_BACKSPACE
Insert	KEY_IC
Delete	KEY_DC
上方向键	KEY_UP
左方向键	KEY_LEFT
F1	KEY_F(1)
F2	KEY_F(2)
PageUp	KEY_PPAGE
PageDown	KEY_NPAGE

TAB、ESC、ENTER 都是ASCII 值对应。

getch() 从stdscr 中读取按键值，wgetch() 从某一窗口中读取按键值。

§7 窗口WINDOW

WINDOW 是NCURSES 管理的基本单元，也是应用的基础，后面要说到的Menus Library 和Forms Library 都要用到WINDOW 来管理其输出。

每个窗口都对应一个数据结构，在创建窗口时NCURSES 会分配这些空间，所以窗口使用完了应该删除。一个窗口的数据结构里记录有窗口的大小，窗口左上角的位置，一个显示缓冲区，当前颜色等等众多数据。

你应该知道了，stdscr 是NCURSES 缺省的窗口，也是启动后的第一个窗口。newwin() 可以建立一个新窗口，其声明是：

```
WINDOW* newwin(int height, int width, int y, int x);
```

其中(y, x) 是窗口左上角的坐标。窗口建立后，如果想要打开功能键的话，应该调用keypad()。我一般是两个一起使用。

一个窗口使用完后，应该删除掉，函数是delwin()，参数是要删除的窗口指针。在删除窗口之前，你也许要清除窗口里的内容，可以用wclear()。但是wclear() 似乎并不使用当前颜色，而只是使窗口黑掉。

box() 是一个有用的函数，可以给一个窗口画一个边框，一般这样用：box(win, 0, 0);。类似的函数还有wborder()。画框时可能要用到一些ASCII 之外的字符，NCURSES 定义了一些字符，以ASC_ 开头，如果终端不支持这些字符，NCURSES 会自动以相似的字符代替。

在窗口里，你就可以输出东西了，`addch()`、`addstr()` 和 `printw()` 几类函数可以随意使用，但一定要记得`wrefresh()`。要注意的是，窗口的`refresh` 是优化了的，它只会刷新被改变的部分，所以效率是比较高的。

当窗口比较多时，为了显示被埋在下面的窗口，可以使用`redrawwin(win)` 来重绘窗口，这个时候`wrefresh()` 并不是那么有效。

`wnoutrefresh(win)` 和`doupdate()` 是很好的一对搭配。当你窗口很多时，对每个窗口直接用`wrefresh()` 是比交耗时的。这时你可以用`wnoutrefresh()` 来更新每个窗口的缓冲区，最后用`doupdate()` 显示。

我并没有使用复杂的很多个窗口，所以在窗口之间的重叠上不是太了解。

你也可以创建一个窗口的子窗口，函数是`subwin()` 和`derwin()`，它们的定义是：

```
WINDOW *subwin(WINDOW *orig, int lines, int cols, int y, int x);
```

```
WINDOW *derwin(WINDOW *orig, int lines, int cols, int y, int x);
```

它们都是在窗口`orig` 中创建一个`lines * cols` 大小的子窗口，它们的区别是前者的`(y, x)` 是相对于屏幕的，而后的`(y, x)` 是相对于窗口`orig` 的。子窗口共享窗口`orig` 的内存。在`wrefresh` 子窗口前，应该调用`touchwin(orig)` 或`touchline(orig)`，否则可能输出不正常。

§8 颜色

实际上在NCURSES 里颜色只是字符属性的一部分，另一种是`highlighting` (高亮)。我更喜欢将它们统称为颜色。通过设置字符的属性和高亮显示，就可以获得一个漂亮界面。

只有在`start_color()` 后，才能使颜色；当然，高亮显示不需要这样。

NCURSES 预定义了八种颜色，以`COLOR_` 开头，有：`COLOR_BLACK`、`COLOR_RED`、`COLOR_GREEN` 等。比较奇怪一点的是，你必须成对使用这些颜色，也就是同时指定前景色和背景色。`init_pair(pair, g, b)` 用来指定一个`COLOR_PAIR` (“色对”) 的，`pair` 是一个1 到`COLOR_PAIRS-1` 之间的值。`COLOR_PAIRS` 指定了色对的最大数目，相应有`COLORS` 指定了颜色的最大数目。

高亮的属性是以`A_` 开头的，有：`A_NORMAL`、`A_REVERSE`、`A_UNDERLINE`、`A_BOLD` 等。但是这些高亮属性与终端有很大关系，例如我基本上只能用到`A_REVERSE`，也就是反色。

`attron()` 和`attroff()` 用来设置字符的属性，之后输出的字符都会具有改变后的属性。例如`attron(A_REVERSE)` 可以打开反色属性，可能的就是黑底白字变成白底黑字。`attron(COLOR_PAIR(n))` 则可以打开色对`n` 所对应的前景和背景。注意在`attron()` 之后，对应地用`attroff()` 来关闭。在`attron()` 的时候，高亮与颜色应该可以通过“或”运算同时使用。

设定色对的一个技巧是使用颜色本身作序号，如：

```
init_pair (COLOR_RED, COLOR_RED, COLOR_BLACK);
```

这样COLOR_PAIR(COLOR_RED) 就很明显了。

chtype 类型可以同时携带字符及其属性。这是addch() 的参数类型。

wchgat() 是一个极有用的函数，它可以改变从当前光标位置开始若干字符的属性，这个在显示菜单或按钮时特别有用。但我在用这个函数时被郁闷了，不知什么原因，这个函数总是不灵。

我现在使用wchgat() 的方法是：通过derwin() 申请一个子窗口，然后用wchgat() 来改变子窗口中字符的属性，然后更新到父窗口。问题真不少，居然mvwchgat() 不能正常工作，所以不得已我只有先wmove() 然后再wchgat()。

chgat() 的参数有四个，第一个是要改变属性的字符的长度，-1 表示到窗口结尾；第二个是属性；第三个是颜色号，注意不要用COLOR_PAIR，直接用色对的序号；第四个参数保留，真接填NULL。它从当前光标改变若干字符的属性后，会返回原光标处。哦，你应该知道chgat() 与wchgat() 的关系吧。

§9 Panel、Menus 和Forms

NCURSES 还有这三个库：Panel Library、Menus Library 和Forms Library。其头文件分别为<panel.h>、<menu.h> 和<form.h>；编译选项分别为-lpanel、-lmenu 和-lform，注意最好放在-lncurses 前。

Panel 不了解；Menus 和Forms 也只是简单抄了例子。

Menus 与Forms 都要先手动创建基本元素，然后有一个“驱动”函数可以用来驱动Menus 或Forms。你要做的就是从wgetch() 接收按键，转化成命令然后通过“驱动”函数来驱动Menus 或Forms。

Menus 主要提供对菜单的支持；Forms 提供一些输入的部件等。

§10 其它

有很多函数我没有用到，下面简单列出。

erase() 与clear() 类似；而clrtobot() 和clrtoeol() 是更小范围的清除。mvwin() 可以用来移动窗口；还有更多函数参考man 3 newwin。

getch() 的功能键里似乎包含了鼠标的按键。

我并不知道有什么函数可以在newwin() 的同时，保存其时的屏幕的显示，在delwin() 后再恢复之。

NCURSES 对层叠窗口的支持不好。

参考文献

- [1] Eric S. Raymond and Zeyd M. Ben-Halim: *Writing Programs with NCURSES*

- [2] Pradeep Padala: *NCURSES Programming HOWTO*
- [3] `man ncurses(3X)`
- [4] <http://poet.cosoft.org.cn/>: 正在翻译 *NCURSES Programming HOWTO*