



DrArgentum 2 дек 2023 в 15:53

Новые проклятия: руководство по ncurses



Простой



11 мин



16K

Программирование*, C++*, *nix*, C*, Разработка под Linux*

Тutorial

Привет, хабр! Я часто пользуюсь библиотеку ncurses и C на Linux. Она простая, быстрая и функциональная.

На ней можно написать рогалик, консольный редактор текста, файловый менеджер и много чего еще.

Ncurses («new curses») — библиотека, написанная на языках **Си** и Ада, предназначенная для управления вводом-выводом на терминал, в числе прочего, библиотека позволяет задавать экранные координаты (в знакоместах) и цвет выводимых символов. Предоставляет программисту уровень абстракции, позволяющий не беспокоиться об аппаратных различиях терминалов и писать переносимый код.

Изначально была библиотека под названием `curses` (название произошло от сокращения «cursor optimization»). Эта библиотека работала напрямую с терминалом, используя API системы, и при этом позволяла разработчикам не заботиться о различных типах терминалов. Библиотека давала возможность приложению перемещать курсор, создавать окна (не `GUI`), управлять цветом, управлять мышью и т.д. без использования зависящих от системы управляющих символов.

`ncurses` — это клон оригинальной `curses` из System V Release 4.0 (SVr4), которая в свою очередь была улучшением классического `curses` из BSD (название произошло от «new curses»)

Но `ncurses` это не только обёртка над зависимыми от системы управляющими символами, это так же и хороший фреймворк для создания красивого пользовательского интерфейса в текстовом режиме. Библиотека содержит функции управления окнами и меню, а так же «библиотеку-сестру» `panel`, управляющую панелями.

Вот пример приложения, основанного на `ncurses` :



+11

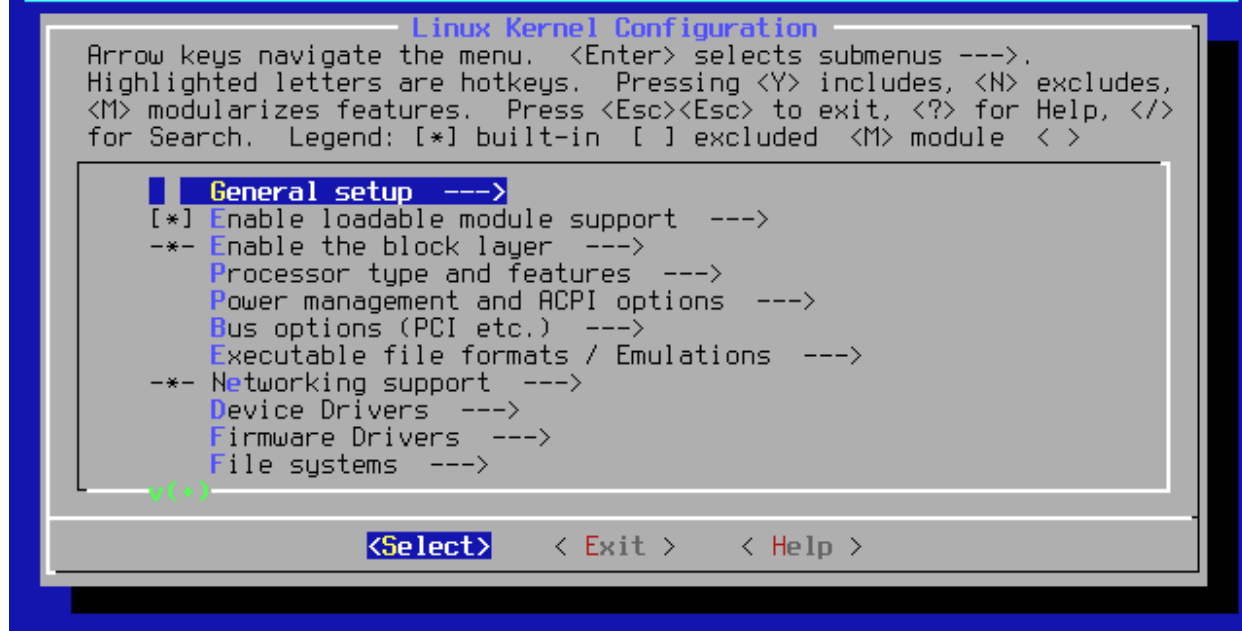


62



9

+9



Программа для конфигурации ядра Linux, написанная на Ncurses

Разумеется, это не все возможности, предоставляемые этой библиотекой.

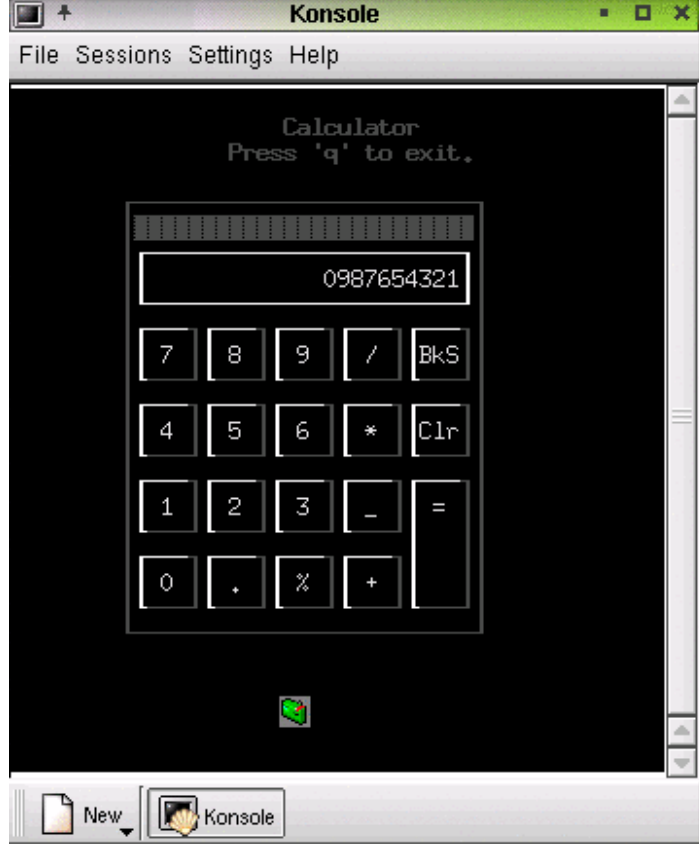
Библиотека ncurses (аналог библиотеки crt в Pascal) предназначена для управления вводом выводом на терминал. Она позволяет программисту не беспокоиться об аппаратных различиях терминалов и писать достаточно переносимый код. Файлы библиотеки находятся в папке /usr/include/ncurses/, дополнительную информацию и примеры можно найти в /usr/share/doc/ncurses-версия_библиотеки и в /usr/share/doc/ncurses-devel-версия_библиотеки.

Библиотека ncurses построена на основе библиотеки curses. Есть и другие библиотеки, построенные на базе curses. Материал представленный здесь построен на общих функциях библиотек ncurses и PDCurses для создания максимально переносимых приложений.

История

В стародавние времена все немногочисленные возможности терминалов (телетайп-терминалов) поддерживались с помощью последовательностей байт. Эти управляющие последовательности, обычно называемые escape -последовательностями, начинались со специального символа 0x1B . Отголоски того времени дошли и до нас. Сегодня мы можем сделать то же самое со своим терминалом (на самом деле с эмулятором терминала, но для краткости, здесь и далее, называемым терминалом), используя точно такую же технологию.

NCURSES (1993) — переписанный старый несвободный терминал CURSES, который был сделан для разработки игры Rogue (1980).



Установка

Установить библиотеку ncurses просто, нам также нужен будет компилятор GCC и система сборки пакетов make.

```
# Arch
sudo pacman -Sy gcc ncurses make

# Debian
sudo apt install libncursesw5-dev gcc make
```

Пример работы

```
#include <ncurses.h> //подключаем библиотеку ncurses

main(int argc, char *argv[])
{
    // инициализация (должна быть выполнена
    // перед использованием ncurses)
    initscr();

    // Измеряем размер экрана в рядах и колонках
    int row, col;
    getmaxyx(stdscr, row, col);
```

```
// перемещение курсора в стандартном экране
move(row / 2, col / 2);

printw("Hello world"); // вывод строки
refresh(); // обновить экран
getch(); // ждём нажатия символа

endwin(); // завершение работы с ncurses
}
```

Что нового Вы увидели в этом примере?

- 1) Чтобы Вы смогли воспользоваться любой библиотекой её нужно подключить. Для библиотеки ncurses это делается добавлением "#include <ncurses.h>" в начале программы.
- 2) Для того чтобы начать работу с библиотекой ncurses необходимо её проинициализировать. Для этого нужно вызвать функцию initscr().
- 3) При завершении работы с библиотекой ncurses нужно вызвать endwin().
- 4) printw и getch аналогичны printf и getchar из <stdio.h>
- 5) библиотека ncurses позволяет ускорить вывод на экран за счёт того, что она использует запись не непосредственно на экран, а в буфер в памяти и когда нужно отобразить все изменения при помощи функции refresh() выводится только изменившаяся часть экрана.

Компиляция

Создаем Makefile со следующим содержимым:

```
# Компиляторы
CC=gcc
CC_FLAGS=-g -Wall -Wl,-O1 -pipe -O2 -flto=2 -fno-fat-lto-objects -fuse-linker-plugin -fPIC

# Цели
TARGET=example

# Директории
SRC_DIR=src
BIN_DIR=bin

# Исходные коды
SOURCES=$(SRC_DIR)/*.c

DEL_FILE      = rm -f
CHK_DIR_EXISTS= test -d
MKDIR         = mkdir -p
COPY          = cp -f
COPY_FILE     = cp -f
COPY_DIR      = cp -f -R
INSTALL_FILE  = install -m 644 -p
```

```
INSTALL_PROGRAM = install -m 755 -p
INSTALL_DIR     = cp -f -R
DEL_FILE        = rm -f
SYMLINK         = ln -f -s
DEL_DIR         = rmdir
MOVE            = mv -f
TAR             = tar -cf
COMPRESS        = gzip -9f
LIBS_DIRS       = -I./include/
LIBS            = $(LIBS_DIRS) -lncursesw
SED             = sed
STRIP           = strip

.PHONY: clean build
all: clean build

build: $(SOURCE)
    $(CC) $(CC_FLAGS) $(LIBS) $(SOURCES) -o $(BIN_DIR)/$(TARGET)

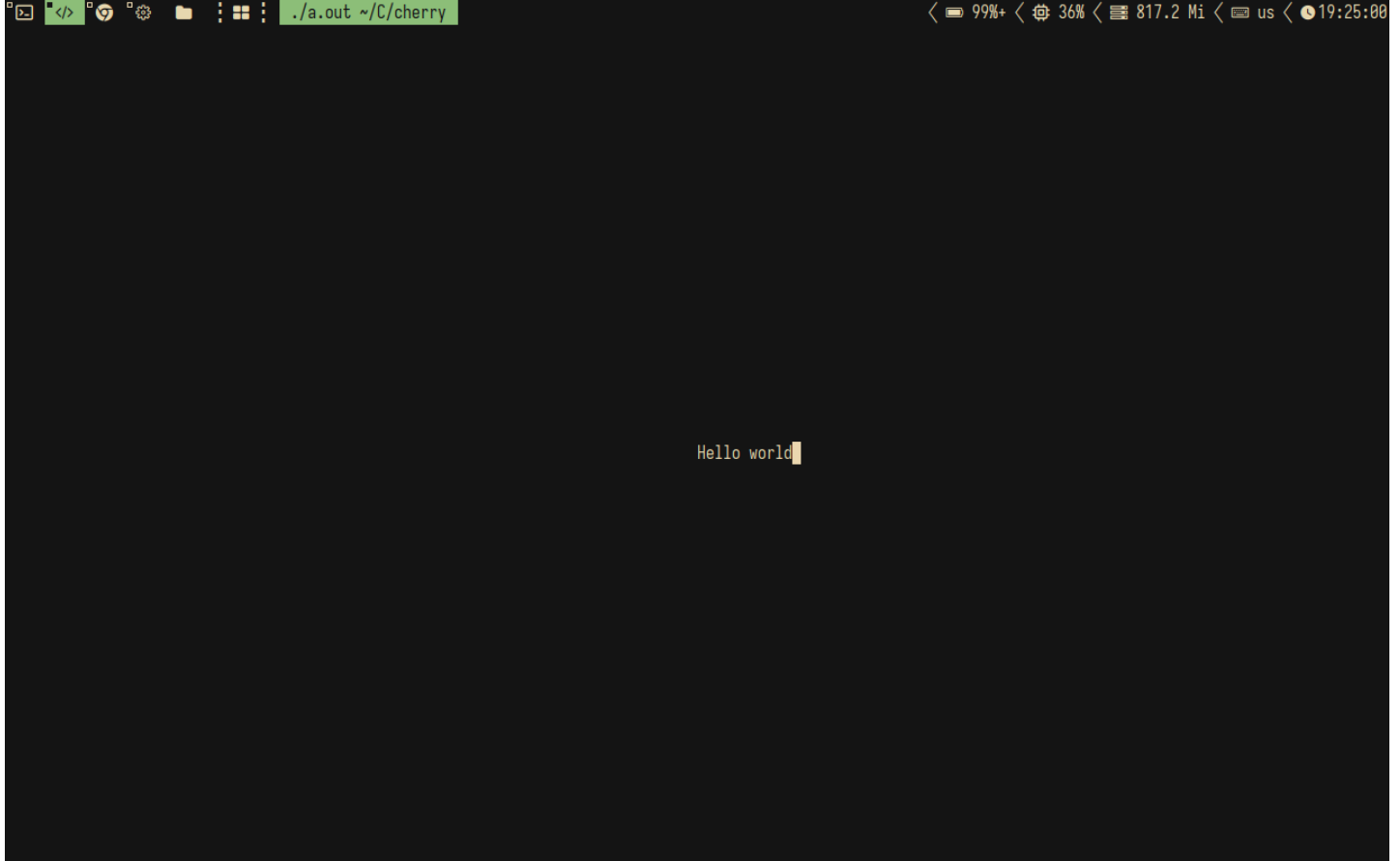
run:
    ./${BIN_DIR}/${TARGET}

clean:
    $(DEL_FILE) bin/*
```

Не буду углубляться в подробности составления Makefile, в сети есть информация об этом

И вводим команду `make` .

На выходе у нас получается следующее:



Мы видим Hello World, который вводится с середине экрана

Вывод символа

`int addch(const chtype ch)`

выводит символ `ch` в текущую позицию курсора и перемещает курсор на один символ вправо или в начало следующей строки, если курсор находится у правой границы экрана. (аналогична функции `putchar` из `<stdio.h>`) **chtype** тип символов с которыми работает `ncurses` (он включает в себя код символа, цвет и дополнительные атрибуты).

Вставка символа

`int insch(chtype ch)`

вставляет символ `ch` слева от курсора и все символы стоящие после курсора перемещаются на одну позицию вправо.

Вывод строки с преобразованием по формату

`int printw(const char *fmt,...)`

пример:

```
...
i=1;
printw("Значение i=%d",i);
...
```

выведет Значение i=1(аналогична функции `printf`)

Вывод строки типа `chtype*`

`int addchnstr(const chtype *chstr, int n)`

выводит первые `n` символов или всю строку символов `chstr`, если `n=-1` в позицию, где расположен курсор.

Вывод строки типа `char*`

`int addstr(const char *str)`

выводит строку `str` в позицию, где расположен курсор.

Вставка строки

`int insnstr(const char* str, int n)`

вставляет первые `n` символов или всю строку символов `str`, если `n=-1` в позицию, где расположен курсор. (положение курсора не изменяется, то что стояло справа от курсора сдвигается вправо)

Вставка пустой строки

`int insertln()`

вставляет пустую строку (строки стоящие ниже начиная с текущей сдвигаются вниз на одну строку)

Режимы вывода

Для символов типа `chtype` можно устанавливать такие атрибуты, как мигание или цвет символа и фона. Для добавления символу атрибута мигания нужно включить флажок `A_BLINK`. Дается это так:

```
chtype ch = 'w' | A_BLINK;
```

Теперь при выводе этого символа он будет мигать, если конечно это позволяет сделать терминал.

(`A_DIM` - пониженная яркость, `A_BOLD` - повышенная яркость, `A_NORMAL` - нормальное отображение, `A_UNDERLINE` - подчёркнутый, `A_REVERSE` - инверсный)

С включением цвета немного сложнее. Перед использованием цветов нужно проинициализировать палитру. Палитра это структура, в которой определённой цифре соответствует определённый цвет. В нашем случае одной цифре соответствуют сразу два цвета символов и фона.

```
...
chtype ch;
...
if (!has_colors())
{
    endwin();
    printf("Цвета не поддерживаются");
    exit(1);
}
start_color();
// 1 цвет в палитре - красные символы на чёрном фоне
init_pair(1, COLOR_RED, COLOR_BLACK);
// 2 цвет в палитре - зелёные символы на желтом фоне
init_pair(2, COLOR_GREEN, COLOR_YELLOW);
```

```
...
ch = 'w' | COLOR_PAIR(1); // символ с цветом 1 из палитры
```

Функция `has_colors` позволяет узнать можно ли использовать цвета. Функция `start_color()` должна вызываться до задания палитры. Функция `init_pair()` нужна чтобы задать какой цифре какой цвет будет соответствовать от 1 до `COLOR_PAIRS-1` (0 зарезервирован для стандартного отображения). Для использования цвета в символе нужно включить флажок `COLOR_PAIR(номер из палитры)`. Список цветов: `COLOR_BLACK`, `COLOR_RED`, `COLOR_GREEN`, `COLOR_YELLOW`, `COLOR_BLUE`, `COLOR_MAGENTA`, `COLOR_CYAN`, `COLOR_WHITE`

Следующие функции позволяют установить атрибуты вывода по умолчанию:

Включение атрибутов

```
int attron(int attrs)
```

включает атрибуты `attrs`. (Например `attron(COLOR_PAIR(1))`; устанавливает цвет 1 из палитры)

Отключение атрибутов

```
int attroff(int attrs)
```

отключает атрибуты `attrs`. (Например `attroff(A_BLINK)`; отключает мигание)

Установка атрибутов

```
int attrset(int attrs)
```

Заменяет текущие атрибуты атрибутами `attrs` (Например `attrset(A_NORMAL)`; заменяет текущие атрибуты на `A_NORMAL`)

Установка атрибутов очистки

```
void bkgdset(chtype ch)
```

Устанавливает атрибуты с которыми очищается экран такими функциями как `clear()`. (Например `bkgdset(COLOR_PAIR(1))`; очистка будет осуществляться цветом 1 из палитры)

Ввод символа

```
int getch()
```

возвращает введенный символ (аналогична функции `getchar`) Если включён режим обработки командных клавиш можно узнать о нажатии функциональных клавиш и клавиш управления курсором. Константы соответствующие кодам этих клавиш можно найти в файле `curses.h`. Вот самые важные из них:

`KEY_DOWN`, `KEY_UP`, `KEY_LEFT`, `KEY_RIGHT` - клавиши стрелок

`KEY_F(n)` Функциональные клавиши `n` может принимать значения от 0 до 63

`KEY_BACKSPACE` Backspace

`KEY_DC` Delete

`KEY_IC` Insert

`KEY_HOME` Home

`KEY_END` End

`KEY_NPAGE` Page Down

`KEY_PPAGE` Page Up

Ввод строки по формату

```
int scanw(char *fmt,...)
```

(аналогична функции scanf) Например для ввода пользователем числа в переменную i вызов функции будет выглядеть так:

```
scanw("%d", &i);
```

Режимы ввода

Автоматическое отображение при вводе

```
int noecho() и int echo()
```

Функция noecho отключает автоматическое отображение при вводе. Это нужно, если программист сам хочет решать вывести ему полученный символ или нет. Функция echo отменяет действие функции noecho.

Установка времени ожидания

```
int halfdelay(int tenths)
```

По умолчанию такие функции как getch ждут ввода до тех пор пока пользователь не нажмёт клавишу. Функция halfdelay устанавливает режим в котором ввод ожидается tenths десятых долей секунды, затем в getch возвращается ERR, если пользователь не нажимал клавиши. Отменить этот режим можно вызвав функцию nodelay().

Обработка командных клавиш

```
int keypad(WINDOW *win, bool bf)
```

По умолчанию обработку таких клавиш, как клавиши управления курсором и функциональные клавиши берёт на себя терминал. Чтобы установить режим обработки командных клавиш нужно вызвать функцию keypad с TRUE в качестве второго параметра. Первый параметр указывает для какого окна Вы хотите установить данный режим. Если Вы не используете окна можно указать stdscr. Для отключения нужно передать FALSE во втором параметре.

Управление курсором

!!! Внимание: Все координаты в ncurses отсчитываются от верхнего левого угла и начиная с 0. Таким образом верхний левый угол имеет координаты (0,0)

Перемещение курсора

```
int move(int y, int x)
```

устанавливает курсор в позицию x,y

Получение текущих координат курсора

```
void getyx(WINDOW *win, int y, int x)
```

В переменные x,y записываются текущие координаты окна win. Если Вы не используете окна в качестве первого параметра можно указать stdscr.

Получить размеры экрана можно вызвав функции getmaxx(stdscr) и getmaxy(stdscr) они возвращают максимальное значение x и y соответственно для данного экрана.

Другие полезные функции

Очистка экрана

`int clear()`

заполняет весь экран пробелами

Очистка от курсора до конца строки

`int clrtoeol()`

заменяет пробелами интервал от курсора до конца строки

Очистка от курсора до конца экрана

`int clrtobot()`

заменяет пробелами интервал от курсора до конца экрана

Вставка/удаление строк

`int insdeln(int n)`

для положительного `n` вставляет `n` пустых строк для отрицательного удаляет `n` строк.

Удаление символа

`int delch()`

удаляет символ на котором стоит курсор (символы стоящие справа от курсора сдвигаются влево)

Удаление строки

`int deleteln()`

удаляет строку на которой стоит курсор (строки стоящие ниже сдвигаются вверх)

Окна

Окно - прямоугольная область экрана с которой можно работать как с целым экраном (очищать, выводить текст и т.д.). То есть осуществлять вывод только в определённую прямоугольную область экрана.

После инициализации создаётся окно `stdscr` с максимально возможными для данного терминала размерами. Если окна не будут использоваться во всех функциях требующих окно можно указывать `stdscr`.

Внутри одного окна также можно создать окна которые будут называться **подокнами**

Создание окна

`WINDOW *newwin(int nlines, int ncols, int begin_y, int begin_x)`

создаёт окно в котором `nlines` строк и `ncols` столбцов (если `nlines` или `ncols` равны 0 то им будет присвоено `begin_y` или `begin_x`) с координатами левого верхнего угла (`begin_x, begin_y`). При вызове с нулевыми аргументами ф-ия создаст окно размером с экран.

Создание подокна

`WINDOW *subwin(WINDOW *orig, int nlines, int ncols, int begin_y, int begin_x)`

создаёт подокно в котором `nlines` строк и `ncols` столбцов (если `nlines` или `ncols` равны 0 то им будет присвоено `begin_y` или `begin_x`) с координатами левого верхнего угла (`begin_x, begin_y`) относительно всего экрана. `origwin` - родительское окно.

Создание подокна 2

`WINDOW *derwin(WINDOW *orig, int nlines, int ncols, int begin_y, int begin_x)`

делает то же самое что и `subwin`, за исключением того, что `begin_x` и `begin_y` координаты относительно окна родителя `origwin`.

Удаление окна/подокна

`int delwin(WINDOW *win)`

удаляет окно/подокно `win`.

Перемещение окна

`int mvwin(WINDOW *win, int y, int x)`

перемещает окно `win` в новую позицию `(x,y)` левого верхнего угла.

Перемещение подокна

`int mvderwin(WINDOW *win, int par_y, int par_x)`

перемещает подокно `win` в новую позицию `(x,y)` левого верхнего угла относительно родительского окна.

Окно играет роль ограничителя вывод на экран. При вводе-выводе в перекрывающиеся окна данные в одном окне могут быть затёрты данными выводимыми в другое окно в месте перекрытия окон. Если Вам нужно обеспечить вывод в перекрывающиеся окна так чтобы вывод в одно окно не повредил данные другого окна можно воспользоваться панелями.

Панели

Под **панелью** в библиотеке `ncurses` понимается окно к которому добавляется глубина. Все панели заносятся в стек панелей. Глубина указывает в каком порядке располагаются панели на экране. Каждую панель можно переместить вверх или вниз всех остальных панелей. Можно скрыть и показать панель.

Для использования панелей нужно подключить библиотеку панелей:

```
#include <panel.h>
```

Если вы используете `panel.h` можно не подключать библиотеку `ncurses.h` потому, что `panel.h` сама подключает `ncurses.h`.

Для компиляции программ использующих панели так же добавляется ключ `-lpanel`:

```
gcc имя_файла.c -o имя_файла -lpanel -lncurses
```

!!! Внимание: при компиляции с библиотеками зависящими от `ncurses` (в данном случае `panel`) ключ подключения `ncurses` (`-lncurses`) должен всегда стоять последним.

Создание панели

`PANEL *new_panel(WINDOW *win)`

Панель создаётся из уже созданного окна `win`. После создания панель перемещается в вершину стека панелей.

Удаление панели

```
int del_panel(PANEL *pan)
```

Удаляет панель pan. Окно связанное с панелью нужно удалять самостоятельно.

Получение окна связанного с панелью

```
WINDOW *panel_window(const PANEL *pan)
```

Возвращает указатель на окно связанное с панелью pan.

Обновление панелей

```
void update_panels()
```

Обновляет стек панелей. Для реального отображения на экране нужно вызвать функцию doupdate().

Скрытие панели

```
int hide_panel(PANEL *pan)
```

Удаляет панель pan из стека панелей. Таким образом при обновлении экрана мы её не увидим.

Отображение панели

```
int show_panel(PANEL *pan)
```

Делает скрытую панель pan снова видимой размещая её при этом поверх остальных панелей.

Перемещение панели поверх всех

```
int top_panel(PANEL *pan)
```

Перемещает панель в стеке поверх всех оставшихся панелей.

Перемещение панели ниже всех

```
int bottom_panel(PANEL *pan)
```

Перемещает панель в стеке ниже всех оставшихся панелей.

Перемещение панели

```
int move_panel(PANEL *pan, int starty, int startx)
```

Перемещает панель pan так, чтобы его верхний левый угол оказался в точке (startx,starty). Положение в стеке панелей не изменяется.

Замена окна

```
int replace_panel(PANEL *pan, WINDOW *window)
```

Заменяет в панели pan текущее окно на окно window. Используется для таких операций как изменение размеров панели. Положение в стеке панелей не изменяется.

Получение панели выше

```
PANEL *panel_above(const PANEL *pan)
```

Позволяет узнать какая панель находится выше панели pan. Если 0 значит выше pan нет панелей.

Получение панели ниже

```
PANEL *panel_below(const PANEL *pan)
```

Позволяет узнать какая панель находится ниже панели pan. Если 0 значит ниже pan нет панелей.

Установка указателя на пользовательские данные

```
int set_panel_userptr(PANEL *pan, const void *ptr)
```

В каждой панели можно установить указатель на любую структуру. Данная функция позволяет установить этот указатель ptr для панели pan.

Получение указателя на пользовательские данные

```
int set_panel_userptr(PANEL *pan, const void *ptr)
```

Данная функция позволяет получить на заданный функцией set_panel_userptr указатель ptr для панели pan.

```
ncdu 1.13 ~ Use the arrow keys to navigate, press ? for help [imported]
--- /mnt/large/memphis/cur/home/zerochan/static -----
      /..
1.6 TiB [██████████] /full
156.4 GiB [          ] /600
39.6 GiB [          ] /240
14.8 GiB [          ]
219.9 MiB [          ]
100.0 KiB [          ]
92.0 KiB [          ]
84.0 KiB [          ]
84.0 KiB [          ]
80.0 KiB [          ]
72.0 KiB [          ]
64.0 KiB [          ]
52.0 KiB [          ]
44.0 KiB [          ]
44.0 KiB [          ] header-7.jpg
40.0 KiB [          ] header.jpg
20.0 KiB [          ] jquery.autocomplete.js
20.0 KiB [          ] asiachan.png
16.0 KiB [          ] jquery.uploadify.js
Total disk usage: 1.8 TiB Apparent size: 1.8 TiB Items: 8856266
```

Item info

Name: full
Path: /mnt/large/memphis/cur/home/zerochan/static
Type: Directory

Disk usage: 1.6 TiB (1,733,686,313,984 B)
Apparent size: 1.6 TiB (1,728,045,757,241 B)

Press i to hide this window

Заклучение

Имея эти знания, вы можете создавать установщики или конфигураторы, редакторы и файловые менеджеры. В общем, единственное ограничение - ваша фантазия.

Полезные материалы (англ)

- [Ncurses Programming Guide](#)
- [Arch man Ncurses](#)
- [Using ncurses library with C](#)
- [Writing programs with ncurses](#)
- [BSD Workshop Ncursess](#)
- [NCURSES Programming HOWTO](#)

Спасибо за прочтение!

Теги: `си`, `ncurses`, `curses`, `linux`, `unix`, `ncurses c`, `C++`, разработка

Хабы: Программирование, C++, *nix, C, Разработка под Linux

Редакторский дайджест



Присылаем лучшие статьи раз в месяц

Электронпочта



8

Карма

44

Рейтинг

Алексеев Бронислав @DrArgentum

Изучаю науку, историю, программирование

Подписаться



Комментарии 9



◉  **includedlibrary** 2 дек 2023 в 17:46


Имхо, лучший обзор ncurses был тут <https://tldp.org/HOWTO/NCURSES-Programming-HOWTO/>
upd:

Но за ссылку https://bsdworks.codeberg.page/c_programming/058_c_programming_ix/ спасибо, это хорошее дополнение к той, что привёл я. В tldp больше справка, а в bsdworks есть ещё и практика



Ответить



◉  **DrArgentum** 2 дек 2023 в 17:48

Добавил

↑  ↓ Ответить  ...

○  **EShin** 2 дек 2023 в 19:05

Ncurses («new curses») — библиотека, написанная на языках **Си** и Ада, предназначенная для управления вводом-выводом на терминал....

Изначально была библиотека под названием `curses` (название произошло от сокращения «cursor optimization»).

`ncurses` — это клон оригинальной `curses` из System V Release 4.0 (SVr4), которая в свою очередь была улучшением классического `curses`

NCURSES (1993) — переписанный старый несвободный терминал **CURSES**, который был сделан для разработки игры **Rogue** (1980).


От автора статьи "Turbo Pascal: 40 лет развития", где 5!!! раз повторяется определение, что такое ЯП Паскаль.

↑  ↓ Ответить  ...

○  **DrArgentum** 2 дек 2023 в 19:08 ^

Ну с кем не бывает, я думаю, что так можно раскрыть всю его суть

↑  ↓ Ответить  ...

○  **atd** 2 дек 2023 в 22:26 ^

Просто ~~контент-форма сломалась~~ тяжело писать по несколько постов в день и ещё их вычитывать...

↑  ↓ Ответить  ...

○  **NeoCode** 2 дек 2023 в 21:27 

Прикольно. Когда-то давно, еще будучи студентом, я сам пытался сделать универсальный текстовый интерфейс. Просто нравилась эта тема, нравилось как выглядят текстовые окна и менюшки:) Применял и в разных лабах/курсовиках, и в каких-то своих программах, да и просто сам процесс разработки нравился. Про `ncurses` тогда не знал, т.к. это линуксовая библиотека, а тогда использовал DOS и ранние Windows.


А сейчас наверное подобная либа, но уже с объектно-ориентированным интерфейсом, пожалуй была бы полезна в Go, учитывая что в основном он применяется на серверах, а там как раз консоли с доступом по ssh.

↑  ↓ Ответить  ...

○  **alex-open-plc** 3 дек 2023 в 10:01 ^

Был и для дося...

<https://www.myabandonware.com/game/curses-31u>

o  **saidelman** 3 дек 2023 в 13:46 ^

Не умею в го, но знаю пару библиотек для TUI:

- <https://github.com/jesseduffield/lazycore>
- <https://charm.sh>

o  **Tuxman** 31 дек 2023 в 23:02

В Ncurses нет виджетов, чтобы начать сразу программировать TUI, т.е. все эти окошечки с бордюрами, прокрутки, всё это самому придётся отрисосывать и события обрабатывать.

Толи дело был Turbo Vision. Сегодня есть современный клон <https://github.com/magiblot/tvision>

На современном C++ я бы попробовал писать TUI на чём-то, типа <https://github.com/ArthurSonzogni/FTXUI>

Зарегистрируйтесь на Хабре, чтобы оставить комментарий

Публикации

ЛУЧШИЕ ЗА СУТКИ ПОХОЖИЕ

 **HireHack** 23 часа назад

«Меня увольняют» или манипуляции HR и менеджеров, чтобы вынудить вас уволиться одним днём. На примере Рольфа

 **Простой**  7 мин  14К

Мнение

 **+102**  65  72 **+72**

 **sobolevn** вчера в 15:17

Как я решил сделать серию видео про устройство интерпретатора CPython

 **Простой**  4 мин  2.1К

Ретроспектива

 +71  65  5 +5



checkpoint 8 часов назад

Обзор приза за первое место в конкурсе «Технотекст 2023»

 **Простой**  5 мин  1.2K

Обзор

 +49  1  12 +12



DAN_SEA 23 часа назад

Самодельная газонокосилка как коммерческий проект?

 **Средний**  11 мин  6.9K

Обзор

 +31  27  22 +22



BabayMazay 3 часа назад

Простой приёмник прямого преобразования для любительской связи на 40, 80 м

 **Средний**  7 мин  976

Тutorial

 +23  7  7 +7



Serverspace 23 часа назад

Как работает графика в видеоиграх?

 **Средний**  9 мин  4.7K

Дайджест

 +23  56  5 +5



vadjuse 23 часа назад

Обратная связь, творящая чудеса

 **Средний**  8 мин  3.2K

 +22  29  20 +20



iviks 1 час назад

Деньги — чужие, проблемы – Ваши

 Простой  3 мин  2.6K

Мнение

 +20  16  28 +28



akostas вчера в 15:06

Как мы создавали собственную дизайн-систему для ускорения процессов разработки

 8 мин  838

Кейс

 +20  29  3 +3



AlexeyNadezhin вчера в 15:02

Батарейки Philips вернулись в Россию

 Простой  2 мин  6.2K

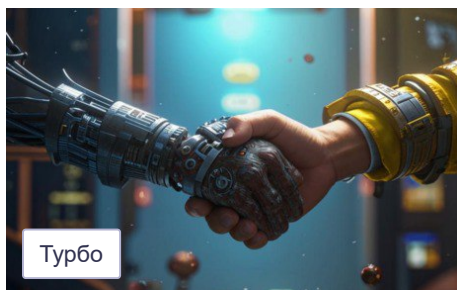
 +19  14  6 +6

Геймифицируй, защищай: как в ЕВРАЗе запустили приложение для охоты на риски

Турбо

Показать еще

МИНУТОЧКУ ВНИМАНИЯ



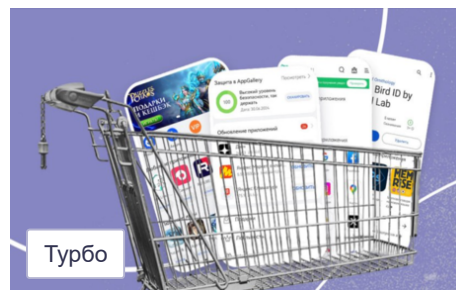
Турбо

Как научить LLM лучше понимать человека



Промо

Есть подозрение, что вам недоплачивают?



Турбо

Оплата смартфоном через NFC в России: миф или реальность?

C / C++ developer (DDos)

от 200 000 ₽ · EdgeЦентр · Можно удаленно

Software Engineer (C/C++, CDN)

от 200 000 ₽ · EdgeЦентр · Можно удаленно

Программист C/C++

от 100 000 ₽ · АО «ТРЭИ» · Пенза

Разработчик C под Linux

до 250 000 ₽ · Averina.agency · Можно удаленно

C developer (алгоритмист)

от 350 000 ₽ · СберТех · Москва

Больше вакансий на Хабр Карьере

ЧИТАЮТ СЕЙЧАС

Сбой в Windows на ПК и серверах после обновления CrowdStrike затронул IT-инфраструктуру компаний, банков и аэропортов

 63K

 304 **+304**

Цифровой апокалипсис сегодня

7.1K 36 +36

Деньги — чужие, проблемы – Ваши

2.7K 28 +28

Глава CrowdStrike Джордж Куртц: хосты Mac и Linux не затронуты, дефект был в коде обновления контента для хостов Windows

2.9K 7 +7

«Меня увольняют» или манипуляции HR и менеджеров, чтобы вынудить вас уволиться одним днём. На примере Рольфа

14K 73 +73

Геймифицируй, защищай: как в ЕВРАЗе запустили приложение для охоты на риски

Турбо

ИСТОРИИ

Опрос: почему студенты бросают IT-курсы?

Онлайн-курсы — один из путей в IT, но не все проходят его до конца. Многие студенты забрасывают учебу. Но почему? Мы решили узнать причины у самих студентов — бывших и нынешних.

Почему люди бросают курсы

Хабр Карьера • Кем работать в IT

Кем работать в IT: мантейнер

О своём опыте работы рассказали ребята из команды базовой операционной системы ОС Astra Linux «Группы Астра»

Кем работать в IT: мантейнер

Хабр Карьера • Активность найма

Активность найма на IT-рынке во 2 квартале 2024

Активность найма во 2 квартале 2024

Хабр Карьера

Зарплаты айтишников во первой половине 2024

Зарплаты в IT во первой половине 2024

ТЕХНОТЕКСТ

По следам «Технотекста»

На Хабре завершился конкурс технических статей «Технотекст 2023». Собрали популярные статьи, победившие в разных номинациях.

Победители Технотекста 2023

Что п...
Недель...
из блог...

Годные компании

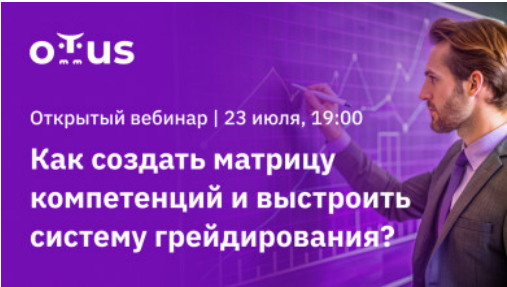
РАБОТА

QT разработчик
8 вакансий

Программист C++
120 вакансий

Программист C
30 вакансий

БЛИЖАЙШИЕ СОБЫТИЯ



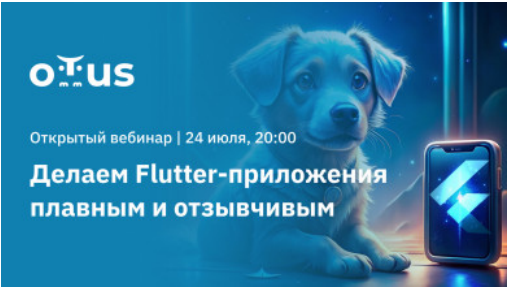
23 июля

Вебинар «Как создать матрицу компетенций и выстроить систему грейдирования»

Онлайн

Менеджмент Тестирование

[Больше событий в календаре](#)



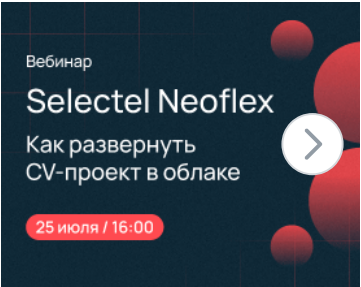
24 июля

Вебинар «Делаем Flutter-приложения плавными и отзывчивыми»

Онлайн

Разработка

[Больше событий в календаре](#)



25 июля

Вебинар «Как пост MLOps-конвейер д проекта в облаке»

Онлайн

Разработка

Администрирование А

[Больше событий в календаре](#)

Ваш аккаунт

Войти

Регистрация

Разделы

Статьи

Новости

Хабы

Компании

Авторы

Песочница

Информация

Устройство сайта

Для авторов

Для компаний

Документы

Соглашение

Конфиденциальность

Услуги

Корпоративный блог

Медийная реклама

Нативные проекты

Образовательные программы

Стартапам



[Настройка языка](#)

[Техническая поддержка](#)

© 2006–2024, Habr