Кроссплатформенная библиотека для работы с текстовым терминалом. Версия руководства 0.0.

Щекин Ярослав

13 июня 2006 г.

1 Общее описание.

Данная библиотека предназначена для создания текстовых интерфейсов пользователя. Предоставляются низкоуровневые функции для работы с терминалом и библиотека widgets.

Поддерживаемые платформы: Все, где работают ncurses (тестирована на Linux), DOS с DJGPP, Windows 9x, Windows NT-based (2000, XP).

2 Установка.

Засада №1. Перед компиляцией необходимо перекодировать исходники в формат данной платформы.

Указать в scr_cnf.h используемую платформу. Скомпилировать. Для использования в своих проектах требуются заголовочные файлы и описываемые модули.

3 Использование в своих проектах.

Для использования базовой (низкоуровневой) функциональности необходимо включать заголовочные файлы "scr_ind.h"и "scr_drv.h", и линковать соответствущие им модули scr_ind.o и scr_drv.o.

Для использования widget-ов необходимо включать "scr_ml.h"и линковать scr_ml.o.

Также можно использовать widget «специальная таблица» (scr_table).

3.1 Особенности использования на различных платформах.

На платформе Linux необходимо линковать с libncurses (gcc ... -lncurses). Также, все определения типов и функций из ncurses будут видимы в использующих библиотеку проектах!

4 Принципы

Экранные координаты начинаются с 0,0 (левый верхний угол).

Координаты всегда задаются в виде (X,Y).

Существует текущая точка вывода (туда будут выводить ф-ии scr_printf, scr_addch), она сдвигается после вывода на позицию после выведенного.

Есть положение терминального (экранного/hardware) курсора. Оно *не зависит* от точки вывода.

Все функции вывода работают *не* с экраном терминала, а с его копией! Для того, чтобы вывести копию на экран, необходимо вызвать scr_refresh.

Widgets *никогда* не вызывают scr_refresh.

Все widgets, которые могут обрабатывать ввод с терминала (клавиатуры), используют для этого функцию scr_{widget}_inject. Её второй параметр — это введённый символ.

Widgets создаются функцией scr_{widget}_create.

5 Описание низкоуровневых функций.

5.1 Инициализация и завершение работы.

```
void scr_initscr(void);
void scr_stopscr(void);
```

Перед началом работы с функциями библиотеки необходимо вызвать scr_initscr, а по завершении — scr_stopscr.

5.2 Вывод на терминал.

```
void scr_cls(void);
void scr_addch(scr_char ch);
void scr_mvaddch(int x, int y, scr_char ch);
void scr_printf(char *format, ...);
void scr_mvprintf(unsigned char x, unsigned char y, char* format, ...);
```

Функция scr_cls очищает экран и устанавливает точку вывода и курсор в верхний левый угол экрана.

Функции scr_(mv)addch выводят один символ типа scr_char в текущую или указанную позицию, а scr_(mv)printf аналогичны обычному printf, и выводят заданную строку с текущим значением экранных атрибутов.

5.3 Ввод символов с «экрана».

```
scr_char scr_inch(void);
scr_char scr_mvinch(int x, int y);
```

Аналогичны соответствущим функциям вывода, но осуществляют ввод с экрана.

5.4 Получение и установка параметров терминала и библиотеки.

```
void scr_locate(unsigned char x,unsigned char y);
void scr_setattr(scr_attr attr);
scr_attr scr_getattr(void);
unsigned char scr_maxx(void);
unsigned char scr_maxy(void);
```

Функция scr_locate устанавливает точку вывода в указанную позицию, scr_(set|get)attr — установка и получение текущих экранных атрибутов, scr_max(x|y) возвращают ширину и высоту экрана.

5.5 Работа с экранными областями.

```
scr_char* scr_getarea(unsigned char x, unsigned char y,
           unsigned char lenx, unsigned char leny);
void scr_regetarea(unsigned char x, unsigned char y,
         unsigned char lenx, unsigned char leny, scr_char* area);
void scr_putarea(unsigned char x, unsigned char y,
           unsigned char lenx, unsigned char leny, scr_char* area);
void scr_freearea(scr_char* area);
```

Функция scr_getarea выделяет область в памяти и копирует туда прямоугольный фрагмент *текущей копии* экрана. scr_regetarea копирует в ранее выделенную область памяти участок экрана, площадь которого должна совпадать с таковой ранее выделенного. scr_putarea выводит ранее запомненный фрагмент по указанным координатам, при этом длина и ширина должны совпадать с ранее использованными. scr_freearea освобождает выделенную ранее функцией scr_getarea память.

5.6 Работа с экранным курсором.

```
void scr_cursor(int x,int y);
void scr_hidecursor(void);
void scr_showcursor(void);
```

Это функции для установки экранного курсора в заданную точку, скрытия или показа его. Перемещение курсора происходит только после обновления!

5.7 Получение клавиатурного/терминального ввода.

```
int scr_getch(void);
```

Ожидает ввода символа (бесконечно), возвращает введённый символ как код в диапазоне 0–256 или одну из констант SCR_KEY_xxx, которые определены в scr_drv.h.

5.8 Преобразования типов.

```
unsigned char scr_char2char(scr_char ch);
scr_attr scr_char2attr(scr_char ch);
scr_char scr_to_scr_char(unsigned char ch, scr_attr attr);
scr_attr scr_color2attr(unsigned int color);
```

Функция scr_char2char возвращает только символ из типа «экранный символ», и так далее. scr_color2attr преобразует цвет в атрибут. См. соответствущие константы в scr_drv.h.