

Введение

Библиотека предназначена для работы с консолью на языке Си. Она очень простая и может использоваться при обучении программированию на Си под Windows. Многие функции аналогичны функциям `conio.h` из Borland Turbo C и модуля CRT из Borland Turbo Pascal. Особенности библиотеки:

- Предназначена для обучения программированию на Си
- Разработана для работы с компилятором Open Watcom C/C++, в перспективе возможна адаптация и для других компиляторов Си
- Снабжена автоматическим инсталлятором
- Часть функций дублирует библиотеки Turbo Pascal и Turbo C
- Проигрывание WAV-файлов
- Работает только под Windows

Установка библиотеки

Для установки библиотеки откройте папку с дистрибутивом и запустите файл `install.bat`. После этого библиотека будет установлена на Ваш компьютер и Вы сможете задействовать её в своих программах, подключая её с помощью директивы `#include <crt.h>`

Пример программы

```
#include <crt.h> // Подключить библиотеку

int main()
{
    // Инициализация окна
    setsize(80,25);          // Размер окна -
                             // - 80 столбцов x 25 строк
    textcolor(WHITE);        // Цвет текста - белый
    textbackground(BLUE);    // Цвет фона
    clrscr();                // Очистить окно консоли
    settitle("EXAMPLE: 'Hello, World!'"); // Установить заголовок
                             // окна консоли

    // Печать строки
    gotoxy(35,12);           // Переместить курсор
    sprintf("Hello, World!"); // Печать строки
    hidecursor();            // Скрыть курсор
    // Завершение работы
    while(!kbhit()); // Ожидать нажатия клавиши
    return 0;
}
```

Описание функций

Базовые функции

```
void clrscr();
```

Очищает окно консоли, заполняя его текущим цветом фона

```
void delay(int ms);
```

Осуществляет задержку длительностью ms миллисекунд

```
void gotoxy(int x, int y);
```

Перемещает курсор в столбец x и строку y. Верхний левый угол имеет координаты (1, 1).

```
void textattr(int attr);
```

Установить текущий текстовый атрибут. В старшем полубайте младшего байта хранится цвет фона, в младшем полубайте младшего байта – цвет текста.

```
void textbackground(int color);
```

Устанавливает текущий цвет фона. Содержимое окна не изменяется, но остальной текст будет выводиться уже на новом фоне и `clrscr()` тоже будет его использовать.

```
void textcolor(int color);
```

Устанавливает текущий цвет текста. Содержимое окна не изменяется, но остальной текст будет выводиться уже новым цветом.

```
int wherex();
```

Возвращает координату x курсора (нумерация – с единицы)

```
int wherey();
```

Возвращает координату y курсора (нумерация – с единицы)

Дополнительные функции

```
void settitle(char *title);
```

Устанавливает заголовок окна консоли

```
void gettitle(char *buffer, int bufsize);
```

Возвращает заголовок окна консоли. `char * buffer` указывает на массив, а `int bufsize` – это размер массива.

```
void setsize(int x, int y);
```

Устанавливает размер окна в строках и столбцах

```
int  gettextattr();
```

Возвращает текущий текстовый атрибут

```
void showcursor();
```

Делает курсор видимым

```
void cursorsize(int size);
```

Устанавливает размер курсора. Аргумент `int size` – процент закрашивания (100 – прямоугольник на всё знакоместо, 1 – едва видимая линия внизу)

```
void hidecursor();
```

Делает курсор невидимым

```
int  sound(char *filename);
```

Проигрывает WAV-файл в фоновом режиме

```
int  soundfg(char *filename);
```

Проигрывает WAV-файл «на переднем плане» - т.е. с приостановкой выполнения программы

```
void nosound();
```

Прерывает фоновое проигрывание WAV-файла

Некоторые функции из `conio.h`

```
int  cprintf( const char *format, ... );
```

Форматный вывод данных. Работает так же, как и `printf`, но учитывает текущий цвет и фон, положение курсора и синхронизирована с функцией `delay`.

```
int  cscanf( const char *format, ... );
```

Форматный ввод данных. Работает так же, как и `scanf`, но учитывает текущий цвет и фон, а также положение курсора.

```
char *cgets( char *buf );
```

Получает строку с клавиатуры в массив (буфер) `buf`.

```
int  cputs( const char *buf );
```

Выводит строку на экран.

```
int  getch( );
```

Считывает символ с клавиатуры, не отображая его на экране. Если клавиша не нажата, то ждёт нажатия клавиши.

```
int  kbhit( );
```

Если была нажата клавиша, то возвращает ненулевое значение, иначе – ноль.

```
int  putch( int c );
```

Выводит символ с кодом `c`.

Примечание: если Вы используете библиотеку `crt.h` для вывода форматированного цветного текста, то настоятельно рекомендуется использовать функциями `cprintf`, `cscanf`, `cgets`, `cputs`, `getch`, `putch`, а не их аналогами из `stdio.h` (`cprintf`, `scanf`, `getchar` и т.п.)

Состав дистрибутива

crt.lib	Библиотека xcrt
crt.h	Заголовочный файл библиотеки xcrt
install.bat	Инсталлятор библиотеки (установка происходит в служебные каталоги Open Watcom)
crt.doc	Документация в формате doc
crt.pdf	Документация в формате pdf

БИБЛИОТЕКА XCRT 1.1

© 2007 Алексей Восков

ЛИЦЕНЗИОННОЕ СОГЛАШЕНИЕ

1. Данная библиотека распространяется "как есть" ("as is"). Автор не несёт ответственность за любые результаты использования или неиспользования этой программы и сопутствующих утилит.
2. Вы можете распространять данный дистрибутив в неизменном виде. Дистрибутив распространяется на бесплатной основе.
3. Дизассемблирование, декомпиляция и внесение изменений в дистрибутив запрещены.
4. Разрешается распространение программ, созданных с использованием данной библиотеки.
5. Автор оставляет за собой право внести изменения в лицензионное соглашение для будущих версий программы.