**Инструкция по работе с библиотекой для создания меню.**

**Создание меню.**

Подключаем к проекту файлы menu.h и menu.c.

Меню или подменю состоят из элементов, которые представляют из себя пункт меню в виде структуры:

typedef struct menu\_item\_t

{

const char \*title;

void(\*function\_action)(void);

struct menu\_item\_t \*submenu;

} menu\_item\_t;

где const char \*title – название пункта, void(\*function\_action)(void) – указатель на функцию которая выполнится при выборе данного пункта, если не требуется никакого действия инициализируйте данное поле NULL, struct menu\_item\_t \*submenu – указатель на массив элементов подменю, если подменю нет, инициализируйте данное поле NULL.

Первое:

Создаём функции для действий в каждом пункте меню, например:

void action\_item\_1() {

printf("\nRun action: Item 1\n");

}

void action\_item\_2() {

printf("\nRun action: Item 2\n");

}

void action\_item\_3() {

printf("\nRun action: Item 3\n");

}

Прототип указателя на функцию при необходимости можно изменить в файле menu.h, по умолчанию он имеет вид void(\*function\_action)(void);

Создаём необходимое количество подменю, например:

// Создание подменю

menu\_item\_t submenu\_items[] = {

{"Subitem 1", action\_item\_3, NULL},

{"Subitem 2", NULL, NULL},

{NULL, NULL, NULL} // Завершающий элемент

};

После этого создаём массив элементов menu\_item\_t, которые будут представлять корневое меню. Например:

// Создание корневого меню

menu\_item\_t root\_menu\_items[] = {

{"Item 1", action\_item\_1, NULL},

{"Item 2", action\_item\_2, submenu\_items},

{"Item 3", NULL, NULL},

{NULL, NULL, NULL} // Завершающий элемент

};

**Каждый список меню или подменю обязательно должен заканчиваться элементом у которого поля инициализированы NULL, например:**

**{NULL, NULL, NULL} это важно !**

Следующий шаг, это создание контекста меню через который и будем работать дальше:

menu\_context\_t ctx;

и вызов функции инициализации меню:

menu\_init(&ctx, root\_menu\_items);

в которую необходимо передать контекст и массив элементов, который представляет корневое меню.

По умолчанию степень вложенности меню ограничена 5, это значение можно изменить в файле menu.h изменив #define STAСK\_SIZE 5

Для обработки пользовательского ввода и передвижения по меню предназначена функция:

void menu\_handle\_input(menu\_context\_t \*ctx, int input);

которая принимает контекст и число от 1 до 4.

1 – вверх, 2 – вниз, 3 – ввод, 4 – возврат.

Всё передвижение по меню вверх или вниз, циклично.

Для вывода пунктов меню куда-либо предназначена функция:

void menu\_display(const menu\_context\_t \*ctx, void(\*display\_callback)(const char\*, int, int));

которая принимает контекст и указатель на пользовательскую функцию для формирования необходимого вида пунктов меню. Прототип указателя на функцию имеет вид:

void(\*display\_callback) (const char\*, int, int);

где первый аргумент — это название пункта, второй аргумент индекс пункта, третий аргумент показывает является ли пункт выбранным или нет.

Пример:

void display\_callback(const char \*title, int index, int is\_selected) {

if (is\_selected) {

printf("> %s\n", title); // Выбранный пункт выделяется символом ">"

} else {

printf(" %s\n", title); // Остальные пункты без выделения

}

} тут мы не задействовали index.

Пример меню для отображения в терминале windows:

#include <stdio.h>

#include <conio.h> // Для функции \_getch() в винде

#include "menu.h"

// Пример функций действий

*void action\_item\_1() {*

*printf("\nRun action: Item 1\n");*

*}*

*void action\_item\_2() {*

*printf("\nRun action: Item 2\n");*

*}*

*void action\_item\_3() {*

*printf("\nRun action: Item 3\n");*

*}*

// Пример Callback-функции для вывода в терминал

*void display\_callback(const char \*title, int index, int is\_selected) {*

*if (is\_selected) {*

*printf("> %s\n", title); // Выбранный пункт выделяется символом ">"*

*} else {*

*printf(" %s\n", title); // Остальные пункты без выделения*

*}*

*}*

// Функция для получения пользовательского ввода

*int get\_user\_input() {*

*char ch = \_getch();* // Считываем нажатую клавишу

*switch (ch) {*

*case 'w': return KEY\_UP;* // Стрелка вверх

*case 's': return KEY\_DOWN;* // Стрелка вниз

*case '\r': return KEY\_ENTER;* // Enter

*case 'b': return KEY\_BACK;* // Back

*default: return -1;* // Некорректный ввод, в мк это можно не добавлять

*}*

*}*

int main() {

// Создание подменю

*menu\_item\_t submenu\_items[] = {*

*{"Subitem 1", action\_item\_3, NULL},*

*{"Subitem 2", NULL, NULL},*

*{NULL, NULL, NULL}* // Завершающий элемент

*};*

// Создание корневого меню

*menu\_item\_t root\_menu\_items[] = {*

*{"Item 1", action\_item\_1, NULL},*

*{"Item 2", action\_item\_2, submenu\_items},*

*{"Item 3", NULL, NULL},*

*{NULL, NULL, NULL}* // Завершающий элемент

*};*

// Инициализация контекста меню

*menu\_context\_t ctx;*

*menu\_init(&ctx, root\_menu\_items);*

*printf("=== Menu is run ===\n");*

*printf("Use:\n");*

*printf("'w' - Up, 's' - Down, Enter - Set, 'b' - Back\n");*

*while (1) {*

// Отображение текущего меню

*menu\_display(&ctx, display\_callback);*

// Получение пользовательского ввода

*int input = get\_user\_input();*

*if (input == -1) {* //в мк это не нужно, так для примера

*printf("\nIncorrect input. Try again.\n");*

*continue;*

*}*

// Обработка ввода

*menu\_handle\_input(&ctx, input)*

*printf("\n");*

*}*

*return 0;*

*}*