C: ?????

Исходный код из книги по C

P. J. Plauger’s The Standard C Library - желательная для прочтения.

Поиск нужного пакета:

```
apt-cache search manpages | grep ^manpag
```

Установка пакетов

```
sudo apt install manpages-ru manpages-ru-dev
```

Список доступных локалей

```
locale -a
```

Установка локали для man страницы

```
LANG=ru_RU.utf8 man 2 open
```

Компиляция

```
cc test.c
cc -0 test test.c
```

Вариант через gcc

```
gcc -Wall -Wextra -Werror -g program.c -o my_program
```

Добавить поиск в текущей директории (чтобы не писать ./)

```
export PATH=$PATH:.
```

Базовое описание языка

```
1 // A first program in C.
2 #include <stdio.h>
3
```

```
// function main begins program execution
int main(void) {
   printf("Welcome to C!\n");
}
```

- директивы препроцессора.

*.h заголовочные файлы, в которых содержится описание функций и ссылки на файлы с их расположением.

Константы

добавляется слово const

```
const int var=7;
```

Комментарии

```
// comments
/* comments
long */
```

Циклы

Цикл for:

```
for (j=0; j<9; j++) {}
```

Цикл с предусловием:

```
while (i < 5) {}
```

Цикл с послеусловием:

```
do {
    // тело цикла
} while (условие);
```

break — немедленный выход из цикла
continue — переход к следующей итерации (пропуск остатка тела)

Функции

Шаблон описания

```
тип_возвращаемого_значения название(входящие аргументы) {

}

int add(int a, int b);     // Функция, возвращающая int
void print_message();      // Функция, не возвращающая значение
int (*func_ptr)(int, int); // Указатель на функцию
```

void - отсутствие аргумента

Функция int main(void) - точка входа.

Получение нескольких значений из функции

Передача указателя

```
#include <stdio.h>

// Функция возвращает int, а через указатели изменяет float
int get_values(int *integer_part, float *float_part) {
    *integer_part = 42;      // Записываем по адресу
    *float_part = 3.14;      // Записываем по адресу
    return 1;                // Статус: успешно
}

int main() {
    int num;
    float pi;

    int status = get_values(&num, &pi);  // Передаем адреса!

    printf("Целое: %d\n", num);   // 42
    printf("Дробное: %f\n", pi);  // 3.140000
    printf("Статус: %d\n", status); // 1

    return 0;
}
```

Структура

```
#include <stdio.h>

// Определяем структуру для хранения двух значений
typedef struct {
    int integer;
    float decimal;
} TwoValues;

// Функция возвращает структуру
TwoValues get_values() {
    TwoValues result;
    result.integer = 42;
    result.decimal = 3.14;
    return result;  // Возвращаем структуру целиком
}

int main() {
    TwoValues values = get_values();

    printf("Целое: %d\n", values.integer);   // 42
    printf("Дробное: %f\n", values.decimal); // 3.140000

    return 0;
}
```

Глобальные переменные

Массив

```
#include <stdio.h>

void get_values(int arr[]) {
    arr[0] = 42;
    arr[1] = 100;
    arr[2] = 255;
}

int main() {
    int values[3];
    get_values(values);  // Массив передается по указателю
```

```
    for (int i = 0; i < 3; i++) {
        printf("%d ", values[i]);  // 42 100 255
    }

    return 0;
}
```

Revision #18
Created 5 July 2026 05:36:31 by Admin
Updated 18 July 2026 14:52:08 by Admin