

Базовые понятия языка Си

Рассматриваемый материал

- Директива предпроцессора `#include`
- Функция `main`
- функция `printf` и форматированный вывод
- типы данных
- переменная и ее адрес
- ввод данных

Директива препроцессора `#include`

Директивы препроцессора - это такие команды которые будут выполняться не просто до запуска программы, но даже до компиляции. В директивы препроцессора подключаются внешние заголовочные файлы и определяются некоторые абсолютные значения проекта.

Обратите внимание, что директивы препроцессора это достаточно сложный инструмент. Использовать его только для определения константных значений не лучшее архитектурное решение.

Для нашего проекта нам понадобится директива `include <stdio.h>`. Эта директива подключит библиотеку стандартного ввода вывода в наш проект.

```
#include <stdio.h>
```

Функция main

Функция main – это точка входа в программу. Поскольку программы написанные на языке си работают на одном уровне с операционной системой, а не в среде виртуализации, то программы должны сообщать операционной системе, что они отработали нормально. Это делается посредством возврата в качестве результата работы программы кода «0». Поэтому тип возвращаемого значения функции Main integer. Любой другой возвратный код сообщает системе, что программа завершена некорректно и необходимо освободить занятые ей ресурсы.

```
int main (int argc, const char* argv[]) {  
    return 0;  
}
```

Функция printf

Общение с пользователем в чистом Си происходит через консоль. Для того чтобы выводить какую-либо информацию используется функция printf предназначенная для форматированного вывода некоторого текста в консоль. Функция описана в заголовке `stdio.h`, поэтому мы и включили данный заголовок в нашу программу.

Какого рода форматирования применяется при выводе строк в консоль?

Существуют два основных инструмента придания выводу необходимого вида:

- экранированные последовательности
- заполнители

Экранированная последовательность

Экранированная последовательность – это буква или символ написанные после знака «\» и при выполнении программы будут на что то заменены. Самые часто используемые – это «\n» «\t» «\\» «\0»

Пример

```
printf("Hello\0 World! \n");  
printf("This is a new \trow with tab. ");  
printf("This is a new row with \\ symbol");
```

Результатом выполнения программы

```
Hello World!  
This is a new   row with tab. This is a new row with \ symbol
```

Заполнитель

Заполнитель – тоже специальная последовательность, но она говорит компилятору, что на место этой последовательности необходимо вставить некий аргумент, который будет передан после строки. Заполнитель начинается со знака процента и обозначает тип вставляемой переменной. Часто используемые:

`%d %s %c %p %lf %x %%`

Пример

```
int a = 50;  
printf("text and%5d%%", a);  
return 0;
```

Результат выполнения программы

text and 50%

Типы данных

| | |
|-----------------|--|
| | |
| short | 2 bytes -32768...32767 |
| int | 4 bytes -2.147.000000 ...+2.147.000000 |
| Unsigned int | 0...+4.294.000000 |
| char | 1 byte -128...+127 |
| Unsigned char | 0...255 |
| Float | 4 bytes |
| double | 8 bytes |
| No boolean type | True=1 false=0 |

Переменная и ее адрес

Переменная это именованный контейнер. У каждого такого контейнера есть адрес в памяти. Язык Си позволяет взять этот адрес и работать с ним. Оператор взятия адреса это знак «&» поставленный перед именем переменной. Т.е. у любой переменной всегда есть значение и адрес, где хранится данное значение.

Пример

```
printf("Переменная a имеет значение: %d \n", a);  
printf("Переменная a хранится по адресу: %p \n", &a);
```

Результат выполнения программы

Переменная a хранится по адресу: 0x7fff5dc8252c

Ввод данных

Одним из способов ввода является функция `scanf`. Принцип ее работы очень схож с функцией `printf`.

Пример

```
int input;
printf("Введите, пожалуйста, число:");
scanf("%d", &input);
printf("Вы ввели число %d, мы удвоили его для Вас: %d\n", input, input*2);
```

Результат работы программы

Введите, пожалуйста, число:321

Вы ввели число 321, мы удвоили его для Вас: 642