

Зачем нужны указатели в языке Си?

В блоге уже есть урок, рассказывающий об указателях. В том уроке я попытался подробно объяснить, что такое указатель и как с ним можно работать. Но забыл самое главное, то с чего стоило бы начать. И вдумчивые читатели быстро заметили этот мой недочет, и стали задавать логичные вопросы: «А зачем это всё? Зачем нужны указатели? Почему нам не работается с переменными? Нормально же всё было, ну чего ты начал-то? А?»

И правда, чего я начал-то? Пора расставить точки над i. Итак, ниже несколько вариантов ответа на вопрос: «Зачем в Си нужны указатели?»

Ответ номер один.

Правила языка Си таковы, что без использования указателей некоторые вещи невозможно реализовать.

Умные слова хорошо, а наглядный пример еще лучше.

Напоминаю, что в языке Си переменные в функцию передаются «по значению». Это обозначает, что когда мы вызываем функцию с какими-то параметрами, то в памяти создаются копии этих переменных и уже с ними работает функция. Когда функция заканчивает свою работу, эти копии уничтожаются.

И чем это нам мешает?

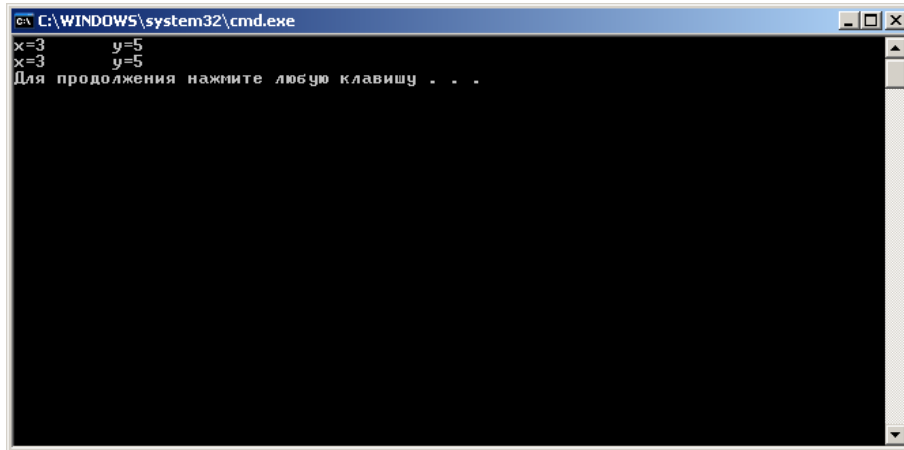
Ну, в принципе, тем, что мы не можем написать функцию, которая меняет значения двух переменных местами. В принципе, я уже говорил об этой проблеме и даже приводил её решение, но как-то не акцентировал внимания на том, что без указателей написать такую функцию невозможно. Но чтобы всё было в одном месте, кратко повторюсь.

На первый взгляд кажется, что всё в порядке, но запустив следующую программу, легко убедиться, что написанная нами функция работает не так, как мы ожидаем.

```
#include <stdio.h>
void swap(int a, int b) {
    int temp = a;
    a = b;
    b = temp;
}

int main() {
    int x = 3, y = 5;
    printf("x=%d \t y=%d \n", x, y);
    swap(x, y);
    printf("x=%d \t y=%d \n", x, y);
    return (0);
}
```

А ниже результат её работы.



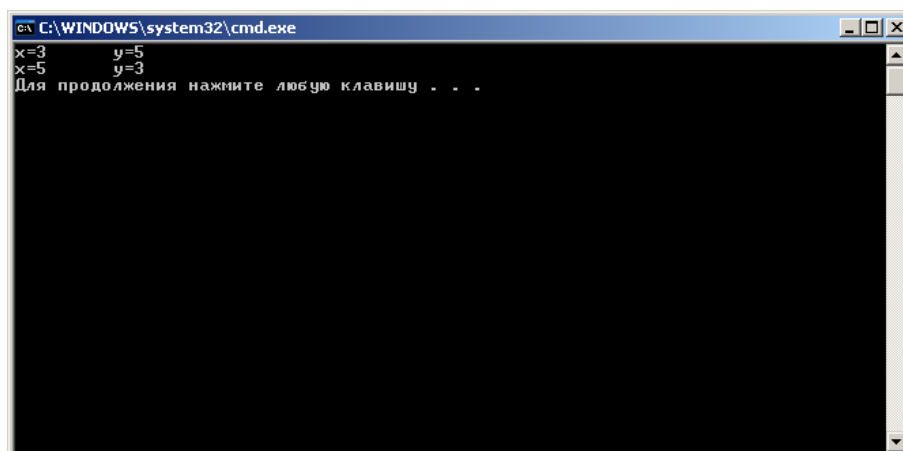
```
C:\WINDOWS\system32\cmd.exe
x=3      y=5
x=3      y=5
Для продолжения нажмите любую клавишу . . .
```

Решение простое, использовать указатели. Т.е. передавать в функцию не копию переменной, а её адрес в памяти и работать уже непосредственно с этим адресом. Такой способ передачи аргументов в функцию, называется передача аргументов «по ссылке».

```
#include <stdio.h>
void swap (int *pi_a, int *pi_b){
    //принимаем указатели на переменные типа int
    int temp = *pi_a;
    *pi_a = *pi_b;
    *pi_b = temp;
}

int main (){
int x=3,y=5;
    printf("x=%d \t y=%d \n",x,y);
    // ВНИМАНИЕ! передаем адреса,
    //так как функция swap принимает указатели
    swap(&x,&y);
    printf("x=%d \t y=%d \n",x,y);
    return(0);
}
```

Результат работы программы с использованием указателей:



```
C:\WINDOWS\system32\cmd.exe
x=3      y=5
x=5      y=3
Для продолжения нажмите любую клавишу . . .
```

Как видите, теперь функция работает так, как от неё требуется.

Ответ номер два.

Указатели позволяет эффективно использовать возможности и ресурсы компьютера.

Хотя ответ и другой, а причина всё та же. Передача аргументов в функцию. Я уже говорил выше, в функцию передаются копии переменных. Это правило не касается массивов. Массивы всегда передаются «по ссылке». Именно это и позволяет эффективно использовать ресурсы и возможности компьютера.

Представьте, что вы бы передавали массив из 10000 значений типа double по значению. То есть, при вызове функции программе нужно было бы найти достаточно места, чтобы сохранить такой объем данных. Причем места непрерывного, так как массивы, как вы должны помнить, располагаются в памяти последовательно. И, кроме того, необходимо было бы скопировать все элементы из одного массива в другой.

Такой подход занимал бы уйму времени и памяти, да и вообще, был бы невозможен при больших массивах. А так, мы передали указатель на первый элемент и радуемся. И не нужно много памяти искать, и копировать ничего не приходится.

Третий ответ.

Использование указателей позволяет писать эффективные программы.

Пока что поверьте на слово. Я знаю один показательный пример, который добавлю сюда чуть позже.

Четвертый ответ.

Это ответ скорее относится к вопросу: «Зачем вам нужно знать указатели?», а не к изначальному вопросу: «Зачем указатели нужны в Си, в принципе?».

Ответ банальный и очевидный.

Язык Си устроен таким образом, что, не зная принципов работы с указателями, вы просто не сумеете использовать огромное количество его возможностей.

Пример привести очень легко. Откройте справочник по функциям работы со строками в Си и внимательно посмотрите, что возвращают эти функции. Огромное их число возвращают в качестве результат своей работы указатели. А если вы о них ничего не знаете, как вы сможете их использовать? Да никак.

Вот и всё на сегодня.