

Программирование на языке C



Модуль 9.

АДРЕСНАЯ АРИФМЕТИКА

- Декларация указателей
- Операции с указателями
- Использование указателей как аргументов функции
- Указатели на функции
- Указатели и массивы
- Указатели и строки



Понятие указателя. Определение указателей на объекты

- **Указатель** — переменная простого производного типа данных, значением которой выступает *адресное выражение*, или *указатель-константа*. Значением указателя, как правило, служит беззнаковый целочисленный адрес другой переменной (ее первого байта), функции (точки входа в нее) либо не равный никакому другому условно «нулевой» адрес, обычно обозначаемый неарифметической константой NULL
- **Определение указателя** на переменную конкретного типа данных
*<имя типа> * <идентификатор>;*
- **Определение указателя** на переменную произвольного типа данных
void * <идентификатор>;

Операции над указателями на объекты

- **Важнейшими операциями над указателями на объекты являются:**

- присваивание значения (=);
- разыменование (косвенная адресация) (*);
- получение адреса указателя (&);
- изменение значения указателя (++ , --);
- операции типа «сложение» (адресная арифметика, см. ниже);
- операции сравнения с указателями на объекты того же типа или константой NULL (<, <=, >, >=, ==, !=)

- **Допустимые арифметические операции над указателями на объекты (L — левый операнд, R — правый):**

- сложение без знака (+) указателя (L) с целочисленным выражением (R);
- вычитание без знака (-) целочисленного выражения (R) из указателя(L);
- вычитание со знаком (-) указателей на объекты одного типа (L, R)

Практика

- Написать функцию обмена значениями для двух переменных.
- Написать функцию вычисления минимума и максимума для одномерного массива.



Указатели и массивы

- Согласно синтаксису языка Си идентификатор массива без индексов элементов — это указатель-константа со значением адреса нулевого (имеющего индекс [0]) элемента массива
- С учетом адресной арифметики, обращение к произвольному элементу массива данных любого типа может производиться одним из двух равнозначных способов, которые допускается комбинировать в случае многомерных массивов:

<имя массива>[<индекс элемента>]
**(<имя массива> + <индекс элемента>)*

- **Одномерный массив**

```
int data[5];
```

```
int *p;
```

```
p=data;
```

data[5] это *(data+5) это *(5+data) это 5[data]

p[5] это *(p+5) это *(5+p) это 5[p]

- **Массив массивов**

```
int data2[3][7];
```

```
??? p2;
```

```
p2 = data2;
```

Функция как производный тип. Указатель на функцию (начало)

- Функция как производный тип языка введена в Си для решения задач, в которых функция (ее адрес) должна являться операндом некоторых операций, параметром другой функции или возвращаемым другой функцией результатом
- **Указатель на функцию** — адресное выражение (в частном случае — переменная-указатель), значением которого выступает адрес первого байта (первого машинного слова) исполнимого кода функции
- Значением идентификатора любой функции, введенного в ее определении (прототипе), является константный указатель на эту функцию

Указатель на функцию (продолжение)

- **Определение переменной-указателя на функцию с конкретной спецификацией параметров**

*<тип результата> (* <идентификатор>)
([<спецификация формальных параметров>]);*

- **Три формы записи операции вызова функции ()**

– явный вызов по имени (константному указателю)

<имя функции> (<список фактических параметров>)

– вызов с разыменованием неконстантного указателя

*(* <идентификатор>) (<список фактических параметров>)*

– вызов без разыменования неконстантного указателя

<идентификатор> (<список фактических параметров>)

Указатель на функцию (окончание)

- **Допустимые операции над указателями на функции**
 - присваивание значения (=) (только для константных и переменных указателей на функции того же типа — с совпадающими типами всех формальных параметров и возвращаемого значения);
 - разыменование (косвенная адресация) (*);
 - получение адреса указателя (&);
 - операции сравнения с указателями на функции того же типа (см. выше) или константой NULL (==, !=)
- Арифметические операции над указателями на функции запрещены

Практика / ДЗ

- Сортировка одномерного массива с использованием функции **qsort()**.
- Выполнить сортировку одномерного массива так, чтобы сначала шли все чётные числа по возрастанию, а затем нечётные по убыванию.



Строки

- Строки в Си реализованы как одномерные массивы символов. Последним символом в строке должен быть нулевой байт – ‘\0’.
- Примеры объявления строк:
 - `char * s1 = "Hello";`
 - `char buf[50] = {0};`
 - `char * s2 = malloc(length);`
 - `char txt[20][80];`
 - `char * s3[20];`

Практика / ДЗ

- Написать функцию определения длины строки.
- Написать функцию делающую все буквы в строке заглавными.
- Написать функцию сложения двух строк.



Список литературы

- [Кнут08] Кнут Д.Э. Искусство программирования / Пер. с англ. — Т. 3. Сортировка и поиск. — 2-е изд. — М.: Вильямс, 2008. — 824 с.
- [КР92] Керниган Б., Ритчи Д. Язык программирования Си / Пер. с англ. — М.: Финансы и статистика, 1992. — 272 с.
- [КР06] Керниган Б., Ритчи Д. Язык программирования С / Пер. с англ. — М.: Вильямс, 2006. — 304 с.
- [Под04] Подбельский В.В., Фомин С.С. Программирование на языке Си. – 2-е доп. изд. – М., Финансы и статистика, 2004. – 600 с.
- [Уэз82] Уэзерелл Ч. Этюды для программистов / Пер. с англ. — М.: Мир, 1982. — 288 с.

