Секция
«Методология современных информационных технологий»

УДК 621.762.4

Борсук Д.Н.

БИТОВЫЕ ПОЛЯ

БНТУ, Минск
Научный руководитель: Дробыш А.А.

Би́товое поле (англ. *bit field*) в программировании – это некоторое количество бит, расположенных последовательно в памяти, значение которых процессор не способен прочитать из-за особенностей аппаратной реализации.

В противоположность другим компьютерным языкам, язык С имеет возможность, называемую битовыми полями, позволяющую работать с отдельными битами. Битовые поля полезны по нескольким причинам. Битовые поля применяются для максимальной упаковки информации, если не важна скорость доступа к этой информации. Например, для увеличения пропускной способности канала при хранении, передаче информации по сети или для уменьшения размера информации при хранении. Также использование битовых полей оправдано, если некоторые интерфейсы устройств передают информацию, закодировав биты в один байт, а также, если некоторым процедурам кодирования необходимо получить доступ к отдельным битам в байте. Хотя все эти функции могут выполняться с помощью битовых операторов, битовые поля могут внести большую ясность в программу. Метод использования битовых полей для доступа к битам основан на структурах. Битовые поля должны объявляться как int, unsigned или signed. При объявлении битового поля после его имени указывается длина поля в битах:

```
{
    unsigned  a1: 1;              // 1 бит
    signed  b1: 3;                // 3 бита
    int  c1: 6;                   // 6 бит
    };
```

Битовые поля могут иметь длину от 1 до16 бит для 16-битных сред и от 1 до 32 бит для 32-битных сред. Длина битового поля должна быть неотрицательным целым числом и не должна превышать длины базового типа данных битового поля:

```
struct device {
unsigned active : 1;
unsigned ready : 1;
unsigned xmt_error : 1;
} dev_code;
```

Данная структура определяет три переменные по одному биту каждая. Структурная переменная dev_code может, например, использоваться для декодирования информации из порта ленточного накопителя. Следующий фрагмент кода записывает байт информации на ленту и проверяет на ошибки, используя dev_code, rd() возвращает статус ленточного накопителя wr_to_tape(), записывает данные:

```
void wr_tape(char c)
{
while(!dev_code.ready) rd(&dev_code); /* ждать */
wr_to__tape (c); /* запись байта */
while(dev_code.active) rd(&dev_code); /* ожида-
ние окончания записи информации */
if(dev_code.xmt error) printf("Write Error");
```

Битовые поля имеют некоторые ограничения. Нельзя получить адрес переменной битового поля. Переменные битового поля не могут помещаться в массив. Переходя с компьютера

на компьютер нельзя быть уверенным в порядке изменения битов (слева направо или справа налево). Любой код, использующий битовые поля, зависит от компьютера.

УДК 621.762.4

Веренич М.С.

ИСПОЛЬЗОВАНИЕ ШАБЛОНОВ В C/C++

БНТУ, Минск
Научный руководитель: Дробыш А.А.

Шаблоны – средство языка C++, предназначенное для кодирования обобщённых алгоритмов, без привязки к некоторым параметрам (например, типам данных, размерам буферов, значениям по умолчанию).

В C++ возможно создание шаблонов функций и классов.

Шаблоны позволяют создавать параметризованные классы и функции. Параметром может быть любой тип или значение одного из допустимых типов (целое число, указатель на любой объект с глобально доступным именем, ссылка и т.д.). Например, нам нужен какой-то класс:

```
class SomeClass{
        int SomeValue;
        int SomeArray[20];
        ...
};
```

Для одной конкретной цели мы можем использовать этот класс. Но, вдруг, цель немного изменилась, и нужен еще один класс. Теперь нужно 30 элементов массива `SomeArray` и вещественный тип `SomeValue` элементов `SomeArray`. Тогда мы можем абстрагироваться от конкретных типов и использовать шаблоны с параметрами.