



Объединения

Последнее обновление: 03.06.2017



На структуры во многом похожи объединения. Объединения также хранят набор элементов, но в отличие от структуры все элементы объединения имеют нулевое смещение. А это значит, что разные элементы занимают в памяти один и тот же участок.

Для определения объединений применяется ключевое слово **union** и следующий формальный синтаксис:

```
union имя_объединения
{
    определения_элементов
};
```

Фактически объединение определяется точно также, как и структура, только вместо слова **struct** используется ключевое слово **union**.

Так, создадим простейшее объединение:

```
union code
{
    int digit;
    char letter;
};
```

Объединение code хранит в одном и том же участки памяти объект `int` и объект `char`. Конкретный размер выделенной памяти будет зависеть от системы и реализации, но в общем случае это будет выглядеть примерно следующим образом:



В этом случае объединение `code` на большинстве платформ будет занимать 4 байта. Длина элементов, как здесь, может быть разной, и в этом случае размер объединения вычисляется по наибольшему элементу.

После определения объединения мы можем создать его переменную и присвоить ей какое-либо значение:

```
union code id;
```

При определении переменной объединения мы ее можем сразу инициализировать, но стоит учитывать, что инициализировать мы можем только первый элемент объединения. В данном случае это элемент `digit` типа `int`, поэтому мы можем передать ему только целое число:

```
union code id = {120};
```

Для обращения к элементам объединения, как и в случае со структурами, можно использовать операцию "точка":

```
#include <stdio.h>

typedef union code
{
    int digit;
    char letter;
};

int main(void)
{
    union code id;
    id.digit = 120;
    printf("%d - %c \n", id.digit, id.letter); // 120 - x
    printf("%d - %d \n", id.digit, id.letter); // 120 - 120

    id.letter = 87;
    printf("%d - %c \n", id.digit, id.letter); // 87 - w

    return 0;
}
```

Стоит отметить, что, так как оба элемента занимают одну и ту же память, то изменение одного из них приведет к изменению другого. Так как участок памяти один, и данные фактически одни и те же, только при обращении к `id.digit` данные интерпретируются как объект `int`, а при обращении к `id.letter` – как объект `char`.

С помощью оператора **typedef** можно задать псевдоним для объединения:

```
typedef union code
{
    int digit;
    char letter;
} secret_code;
```

Здесь псевдонимом является идентификатор `secret_code`, поэтому следующие определения переменных будут аналогичны:

```
union code id = {22};
secret_code id2 = {22};
```

Также можно определять анонимные объединения:

```
#include <stdio.h>

union
{
    int digit;
```

```
    char letter;
} id1, id2;    // переменные id1, id2

int main(void)
{
    id1.digit = 122;
    id2.digit = 84;
    printf("%d - %c \n", id1.digit, id1.letter);
    printf("%d - %c \n", id2.digit, id2.letter);

    return 0;
}
```

И как и со структурами, можно определять указатели на объединения. Для обращения к элементам объединения по указателю применяется тот же синтаксис, что и в случае со структурами:

```
(* указатель_на_объединение).имя_элемента
указатель_на_объединение->имя_элемента
```

Используем указатели на объединения:

```
#include <stdio.h>

union code
{
    int digit;
    char letter;
};

int main(void)
{
    union code id = {45};
    union code * p_id = &id;
    printf("%d \n", p_id->digit);    // 45
    p_id->digit= 89;
    printf("%d \n", id.digit);      // 89

    return 0;
}
```

[Назад](#) [Содержание](#) [Вперед](#)



ALSO ON METANIT.COM

Конфигурация и генерация ...

3 месяца назад • 3 comments

Конфигурация и генерация JWT-токенов для аутентификации и ...

Установка веб-сервера Apache и ...

2 месяца назад • 12 comments

Установка веб-сервера Apache и PHP 8 на Mac OS, настройка ...

Множественная регистрация ...

2 месяца назад • 2 comments

Множественная регистрация сервисов в ASP.NET Core и C#, ...

Уязвимс програм

16 дней на:

Уязвимост программы сторонние

5 Comments

metanit.com

 Политика конфиденциальности Disqus Войти ▾ Favorite Твитнуть Поделиться

Лучшее ▾



Присоединиться к обсуждению...

ВОЙТИ С ПОМОЩЬЮ

ИЛИ ЧЕРЕЗ DISQUS **Aslan Almaniyaov** • 4 years ago • edited

Чё то непонятно, зачем такая структура как "объединение" нужна...
И с примером union code { int digit; char letter;}; неясно.

Вот моё предположение: Это могло быть нужно раньше, когда память была ограниченной.
Например, нужно было хранить целые числа и символы, но места не хватало для обоих.
И тогда, понимая, что

Помощь сайту
YooMoney

- 410011174743222

Перевод на карту

- Номер карты: 4048415020898850
- Номер карты: 4890494751804113

[Вконтакте](#) | [Twitter](#) | [Канал сайта на youtube](#) | [Помощь сайту](#)

Контакты для связи: metanit22@mail.ru

Copyright © metanit.com, 2012–2022. Все права защищены.