

🏠 (<http://cppstudio.com>)

/ Стандартные заголовочные файлы из Си в C++ (<http://cppstudio.com/cat/309/>)

/ Заголовочный файл cstdlib (stdlib.h) (<http://cppstudio.com/cat/309/324/>) / Функция realloc

Функция realloc



Оценка: **4,00** (голосов: 1)

Чтобы проголосовать, вы должны зарегистрироваться.

Прототип функции realloc:

```
1 void * realloc( void * ptrmem, size_t size );
```

Название	Язык
stdlib.h	C
cstdlib	C++

Описание

Функция `realloc` выполняет перераспределение блоков памяти.

Размер блока памяти, на который ссылается параметр `ptrmem` изменяется на `size` байтов. Блок памяти может уменьшаться или увеличиваться в размере.

Эта функция может перемещать блок памяти на новое место, в этом случае функция возвращает указатель на новое место в памяти. Содержание блока памяти сохраняется даже если новый блок имеет меньший размер, чем старый. Отбрасываются только те данные, которые не вместились в новый блок. Если новое значение `size` больше старого, то содержимое вновь выделенной памяти будет неопределенным.

В случае, если `ptrmem` равен `NULL`, функция ведет себя именно так, как функция `malloc`, т. е. выделяет память и возвращает указатель на этот участок памяти.

В случае, если `size` равен 0, ранее выделенная память будет освобождена, как если бы была вызвана функция `free`, и возвращается нулевой указатель.

Параметры:

- **ptrmem**

Указатель на блок ранее выделенной памяти функциями `malloc`, `calloc` или `realloc` для перемещения в новое место. Если этот параметр – `NULL`, просто выделяется новый блок, и функция возвращает на него указатель.

- **size**

Новый размер, в байтах, выделяемого блока памяти. Если size равно 0, ранее выделенная память освобождается и функция возвращает нулевой указатель, ptrmem устанавливается в 0.

Возвращаемое значение

Указатель на перераспределенный блок памяти, который может быть либо таким же, как аргумент ptrmem или ссылаться на новое место.

Тип данных возвращаемого значения всегда void*, который может быть приведен к любому другому.

Если функции не удалось выделить требуемый блок памяти, возвращается нулевой указатель, и блок памяти, на который указывает аргумент ptr остается неизменным.

Пример: исходный код программы

```

1 //пример использования функции realloc
2 #include <iostream>
3 #include <cstdlib>
4
5 int main()
6 {
7     int input;
8     int counter = 0; // счетчик введенных чисел
9     int * values = NULL;
10    int * many_numbers;
11
12    do {
13        std::cout << "Введите целое значение (0 - выход): ";
14        std::cin >> input;
15        counter++;
16
17        many_numbers = (int*) realloc (values, counter * sizeof(int)); // при добавлении
18
19        if (many_numbers != NULL)
20        {
21            values = many_numbers;
22            values[counter - 1] = input; // добавить к массиву
23        }
24        else
25        {
26            free (values); // удалить массив
27            std::cout << "Ошибка перевыделения памяти!";
28            exit (1); // завершить работу программы
29        }
30    } while (input != 0); // пока не введен 0
31
32    std::cout << "Введенные числа: ";
33    for (int ix = 0; ix < counter; ix++) std::cout << values[ix] << " ";
34    free (values); // удалить массив
35
36    return 0;
37 }
```

Пример работы программы


Программа предлагает пользователю ввести числа. Каждый раз, когда новое значение сохраняется в блок памяти, он увеличивается на размер int.


CppStudio.com

Введите целое значение (0 – выход): 11
Введите целое значение (0 – выход): 33
Введите целое значение (0 – выход): 55
Введите целое значение (0 – выход): 77
Введите целое значение (0 – выход): 99
Введите целое значение (0 – выход): 0
Введенные числа: 11 33 55 77 99 0

 Обсудить на форуме (/topics/)

 Автор: admin (/forums/users/admin/)

 Дата: 03.09.2012

 Поделиться:

Похожие статьи:

1. Массивы в C++ (<http://cppstudio.com/post/389/>)
2. Функция free (<http://cppstudio.com/post/850/>)
3. Цикл for в C++ (<http://cppstudio.com/post/348/>)
4. Номер min элемента, произведение элементов массива (<http://cppstudio.com/post/1357/>)
5. Аналоги строковых функций C++ (<http://cppstudio.com/post/1381/>)

Комментарии



AlexZ (<https://plus.google.com/u/0/102233245253626250111/>)

01.04.2017 (/post/860/comment-page-1/#comment-3560)

В самом начале в «Описании» серьезная ошибка. Написано: «*Размер блока памяти, изменяется **НА** size байтов*». Как будто size это разница между новым и старым размером. Но на самом деле size это новый размер всего блока.

Дальше в описаниях параметров уже написано правильно.

Войдите, чтобы ответить (http://cppstudio.com/wp-login.php?redirect_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F860%2F)









Оставить комментарий

Вы должны войти (http://cppstudio.com/wp-login.php?redirect_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F860%2F), чтобы оставить комментарий.

Поиск...

Поиск

Translation

 (/post/860/)Русский (/post/860/)
 (/uk/post/860/)Українська (/uk/post/860/)
 (/en/post/860/)English (/en/post/860/)
 (/de/post/860/)Deutsch (/de/post/860/)
 (/be/post/860/)Беларуская (/be/post/860/)
 (/kk/post/860/)Қазақ тілі (/kk/post/860/)
 (/uz/post/860/)O'zbek tili (/uz/post/860/)
 (/tr/post/860/)Türkçe (/tr/post/860/)

Новое

- Особенности Qt: слоты и сигналы, описание QObject и QApplication, виды окон и т.д.
(<http://cppstudio.com/post/11167/>)
- Первая программа на Qt:
(<http://cppstudio.com/post/11127/>)
- Введение – графическая библиотека Qt
(<http://cppstudio.com/post/11097/>)
- Наследование классов
(<http://cppstudio.com/post/10103/>)
- Перегрузка операторов в C++ (часть 2)
(<http://cppstudio.com/post/10058/>)

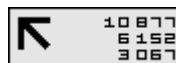
Популярное

Sorry. No data so far.

© 2022 CppStudio – Программирование для начинающих на C++



(<https://plus.google.com/u/0/106109650739084338784>)



(<http://www.liveinternet.ru/click>)



(<http://orphus.ru>)