

[⬆ \(http://cppstudio.com\)](http://cppstudio.com)[/ Стандартные заголовочные файлы из Си в C++ \(http://cppstudio.com/cat/309/\)](http://cppstudio.com/cat/309/)[/ Заголовочный файл cstdlib \(stdlib.h\) \(http://cppstudio.com/cat/309/324/\)](http://cppstudio.com/cat/309/324/) / Функция free

Функция free

Оценка: **4,00** (голосов: 1)

Чтобы проголосовать, вы должны зарегистрироваться.

Прототип функции free:

```
1 void free( void * ptrmem );
```

Название	Язык
stdlib.h	C
cstdlib	C++

Описание

Функция free освобождает место в памяти. Блок памяти, ранее выделенный с помощью вызова malloc, calloc или realloc освобождается. То есть освобожденная память может дальше использоваться программами или ОС.

Обратите внимание, что эта функция оставляет значение ptr неизменным, следовательно, он по-прежнему указывает на тот же блок памяти, а не на нулевой указатель.

Параметры:

- **ptrmem**

Указатель на блок памяти, ранее выделенный функциями malloc, calloc или realloc, которую необходимо высвободить. Если в качестве аргумента передается нулевой указатель, никаких действий не происходит.

Возвращаемое значение

Функция не имеет возвращаемое значение.

Пример: исходный код программы

```
1 //пример использования функции free
2 #include <iostream>
3 #include <cstdlib>
4
5 int main()
6 {
7     int * buffer1 = (int*) malloc(100 * sizeof(int)),           // выделяем память
8         * buffer2 = (int*) calloc(100, sizeof(int)),           // выделяем память
9         * buffer3 = (int*) realloc(buffer2, 500 * sizeof(int)); // перераспредели
10
11     free(buffer1);                                              // высвобождаем б.
12     free(buffer3);                                              // высвобождаем б.
13     return 0;
14 }
```

Пример работы программы

Эта программа не имеет никакого вывода данных на экран. Просто демонстрирует несколько способов выделения и освобождения динамической памяти с помощью функций заголовочного файла `cstdlib`.

Обсудить на форуме (/topics/)

Автор: admin (/forums/users/admin/)

Дата: 02.09.2012

Поделиться:

Похожие статьи:

1. Использование Valgrind для поиска утечек и недопустимого использования памяти (<http://cppstudio.com/post/4348/>)
2. Динамическое выделение памяти в C (<http://cppstudio.com/post/9088/>)
3. Функция realloc (<http://cppstudio.com/post/860/>)
4. Функция calloc (<http://cppstudio.com/post/846/>)
5. Типы данных C++ (<http://cppstudio.com/post/271/>)

Комментарии

Ульян Романов (<http://vk.com/id80710346>)



29.04.2015 (/post/850/comment-page-1/#comment-2704)

То есть мы высвобождаем память сразу и buffer2, и buffer3?

1 **free**(buffer3);

Войдите, чтобы ответить (http://cppstudio.com/wp-login.php?redirect_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F850%2F)



Mercury

11.01.2017 (/post/850/comment-page-1/#comment-3423)

> То есть мы высвобождаем память сразу и buffer2, и buffer3?









На самом деле после вызова функции realloc указатель buffer2, возможно, освобождён и указывает на мусор (а возможно – совпадает с buffer3).

Войдите, чтобы ответить (http://cppstudio.com/wp-login.php?redirect_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F850%2F)

Оставить комментарий

Вы должны войти (http://cppstudio.com/wp-login.php?redirect_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F850%2F), чтобы оставить комментарий.

Translation

 (/post/850/)Русский (/post/850/)
 (/uk/post/850/)Українська (/uk/post/850/)
 (/en/post/850/)English (/en/post/850/)
 (/de/post/850/)Deutsch (/de/post/850/)
 (/be/post/850/)Беларуская (/be/post/850/)
 (/kk/post/850/)Қазақ тілі (/kk/post/850/)
 (/uz/post/850/)O'zbek tili (/uz/post/850/)
 (/tr/post/850/)Türkçe (/tr/post/850/)

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/>)

Популярное

Sorry. No data so far.

➤ Введение – графическая
библиотека Qt
(<http://cppstudio.com/post/11097/>)

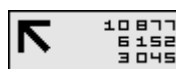
➤ Наследование классов
(<http://cppstudio.com/post/10103/>)

➤ Перегрузка операторов в C++
(часть 2)
(<http://cppstudio.com/post/10058/>)

© 2022 CppStudio – Программирование для начинающих на C++



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



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



(<http://orphus.ru>)