

[⬆ \(http://cppstudio.com\)](http://cppstudio.com)[/ Стандартные заголовочные файлы из Си в C++ \(http://cppstudio.com/cat/309/\)](http://cppstudio.com/cat/309/)[/ Заголовочный файл cstring \(string.h\) \(http://cppstudio.com/cat/309/325/\)](http://cppstudio.com/cat/309/325/) / Функция memmove

# Функция memmove

Оценка: **4,00** ( голосов: 1 )

Чтобы проголосовать, вы должны зарегистрироваться.

## Прототип функции memmove:

```
1 void * memmove( void * destptr, const void * srcptr, size_t num );
```

## Заголовочный файл:

Название	Язык
string.h	C
cstring	C++

## Описание

Переместить блок памяти. Функция копирует num байтов из блока памяти источника, на который ссылается указатель srcptr, в блок памяти назначения, на который указывает указатель destptr. Копирование происходит через промежуточный буфер, что, в свою очередь, не позволяет destination и srcptr пересекаться.

Тип данных объектов, на которые указывают как srcptr так и destptr не имеют никакого значения. Так как эта функция работает с бинарными данными. Функция не проверяет, есть ли символ завершения в srcptr, она всегда копирует количество байтов, указанное в num. Чтобы избежать переполнения блока памяти destptr, размер destptr должен быть не менее num байтов. Однако, может возникнуть ситуация, когда destptr и srcptr пересекутся.

## Параметры:

- **destptr**

Указатель на блок памяти назначения (куда будут копироваться байты данных), имеет тип данных void.

- **srcptr**

Указатель на блок памяти источник (т. е., откуда будут копироваться байты данных), имеет тип

данных void .

- **num**

Количество копируемых байтов.

## Возвращаемое значение

Указатель на блок памяти назначения.

## Пример: исходный код программы

```

1 //пример использования функции memmove
2
3 #include <iostream>
4 #include <cstring>
5
6 int main()
7 {
8     char str[] = "memmove может оказаться очень полезной функцией.....";
9
10    // перемещаем блок символов в памяти
11    memmove (str + 49,                                // та же самая строка, со сдвигом в
12            str + 38,                                // копируем из str начиная с 38-го
13            47);                                       // скопировать нужно 47 байт
14    std::cout << str << std::endl;
15    return 0;
16 }
```

## Пример работы программы

CppStudio.com

memmove может оказаться очень очень полезной функцией...

Обсудить на форуме (/topics/)

Автор: admin (/forums/users/admin/)

Дата: 31.08.2012

Поделиться:

### Похожие статьи:

1. Функция memcpy (<http://cppstudio.com/post/678/>)
2. Функция strncpy (<http://cppstudio.com/post/690/>)
3. string: шаблонный строковый класс STL (<http://cppstudio.com/post/6110/>)
4. Функция strcat (<http://cppstudio.com/post/695/>)
5. Функция strncat (<http://cppstudio.com/post/698/>)

## Комментарии



kpanat . (<https://plus.google.com/100539009002642920953>)

03.11.2015 (/post/682/comment-page-1/#comment-2966)

Ну тут ошибка... написано «Копирование происходит через промежуточный буфер, что, в свою очередь, не позволяет destination и srcptr пересекаться», а надо «Копирование происходит через промежуточный буфер, что, в свою очередь, позволяет destination и srcptr пересекаться» Такая маленькая ошибочка, которая в корне меняет смысл. Но вообще лучше использовать другую функцию для пересекающихся блоков варьируя направлением копирования... Этот параметр не входит ни в одну строковую функцию. но все функции с ним неявно работают устанавливая направление копирования всегда вперёд. Хотя к примеру для корректной вставки необходимо выполнить копирование **назад**... Конечно вам придётся написать свою функцию. В стандарте и расширениях такой нет.

Войдите, чтобы ответить ([http://cppstudio.com/wp-login.php?redirect\\_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F682%2F](http://cppstudio.com/wp-login.php?redirect_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F682%2F))



gam507

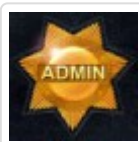
26.12.2013 (/post/682/comment-page-1/#comment-908)

Ошибка в примере. memmove(str+49,str+38,47). Нужно memmove(str+30,str+24,23).

Войдите, чтобы ответить ([http://cppstudio.com/wp-login.php?redirect\\_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F682%2F](http://cppstudio.com/wp-login.php?redirect_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F682%2F))

admin (<http://twitter.com/F45I>)

26.12.2013 (/post/682/comment-page-1/#comment-909)



Для nix-систем тут нет ошибки, так как для хранения русских букв требуется два байта.









Войдите, чтобы ответить ([http://cppstudio.com/wp-login.php?redirect\\_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F682%2F](http://cppstudio.com/wp-login.php?redirect_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F682%2F))

## Оставить комментарий

Вы должны войти ([http://cppstudio.com/wp-login.php?](http://cppstudio.com/wp-login.php?redirect_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F682%2F)

[redirect\\_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F682%2F](http://cppstudio.com/wp-login.php?redirect_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F682%2F)), чтобы оставить комментарий.

## Translation

 (/post/682/)Русский (/post/682/)  
 (/uk/post/682/)Українська (/uk/post/682/)  
 (/en/post/682/)English (/en/post/682/)  
 (/de/post/682/)Deutsch (/de/post/682/)  
 (/be/post/682/)Беларуская (/be/post/682/)  
 (/kk/post/682/)Қазақ тілі (/kk/post/682/)  
 (/uz/post/682/)O'zbek tili (/uz/post/682/)  
 (/tr/post/682/)Türkçe (/tr/post/682/)

## 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/>)

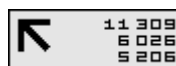
## Популярное

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>)