

[⬆ \(http://cppstudio.com\)](http://cppstudio.com)[/ Стандартные заголовочные файлы из Си в C++ \(http://cppstudio.com/cat/309/\)](http://cppstudio.com/cat/309/)[/ Заголовочный файл cstdio \(stdio.h\) \(http://cppstudio.com/cat/309/323/\)](http://cppstudio.com/cat/309/323/) / Функция setvbuf

# Функция setvbuf

 Оценка: **4,00** ( голосов: 1 )

Чтобы проголосовать, вы должны зарегистрироваться.

## Прототип функции setvbuf:

```
1 int setvbuf( FILE * filestream, char * buffer, int modeopen, size_t size );
```

## Заголовочный файл

Название	Язык
stdio.h	C
cstdio	C++

## Описание

Функция изменяет буфер, который будет использоваться для операций ввода/вывода с указанным потоком. Эта функция позволяет задать режим доступа и размер буфера. Эта функция должна быть вызвана для файла, связанного с потоком, который уже открыт. Причём вызов данной функции должен находиться между операциями открытия и закрытия файла. Размер буфера определяется параметром `size`, в байтах. Если буфер не определен (т.е. параметр `buffer` имеет аргумент `NULL`), система динамически распределяет объем памяти, требуемый функцией `setvbuf` и использует его в качестве буфера для потока. Параметр `modeopen` используется, чтобы определить тип буферизации, т. е. использовать полную буферизацию или не использовать её вообще. С полной буферизацией потоков, операция записи не выполняется непосредственно на физическом устройстве, связанные с ними. Вместо этого, данные накапливаются в буфере и записываются в устройство в качестве блока, когда буфер будет заполнен. Эти данные могут быть принудительно очищены, путём вызова функций `fflush` или `fclose`. Все буферы и обнуляются также, при завершении работы программы. Буфер записывается на физическое устройство каждый раз, когда встречается символ новой строки. С небуферизованных потоков, данные записываются на физическое устройство как можно скорее, после каждой операции записи. Все файлы открываются с буфером, который выделен по умолчанию. Эта функция может быть использована для отключения буферизации или изменения размера буфера для конкретного потока. Системные стандартные потоки, такие, как стандартный вывод и стандартный поток ошибок являются небуферизованными по умолчанию. Параметры:

- **filestream**

Указатель на объект типа `FILE`, который соединён с открытым потоком.

- **buffer**

Выделенный пользователем буфер должен быть размером не менее `size` байт. Если установлено значение `NULL`, функция автоматически выделяет буфер указанного размера.

- **modeopen**

Режимы буферизации файлов:

<code>_IOFBF</code>	<b>Полная буферизация:</b> На выход, данные записываются после заполнения буфера. Входной буфер заполняется при открытии файла и, если буфер пуст.
<code>_IOLBF</code>	<b>Строковая буферизация:</b> На выход, данные записываются, когда символ новой строки вставляется в поток или когда буфер полон. На входе, буфер заполняется до следующего символа новой строки, когда запрашивается операция ввода, при условии, что буфер пуст.
<code>_IONBF</code>	<b>Без буферизации:</b> Нет буфер используется. Каждая операция ввода / вывода написано как можно скорее. В этом случае, <code>thebuffer</code> и размер параметры игнорируются.

- **size**

Размер буфера в байтах. Если аргумент параметра `size` - `NULL`, то автоматически выделяется минимальный размер буфера, в противном случае оно должно быть равным или меньше размера в байтах массива, заданного в качестве буфера.

## Возвращаемое значение

Если буфер корректно указан, возвращается нулевое значение.

В противном случае, возвращается ненулевое значение, которое возникает в результате некорректного указания параметров режима или размера, или, при выделении памяти, если был указан в качестве буфера параметр `NULL`).


## Пример: исходный код программы


```


1  //пример использования функции setvbuf
2
3  #include <iostream>
4  #include <cstdio>
5
6  int main()
7  {
8      FILE *ptrFile = fopen("file.txt", "w");
9
10     // для файла file.txt устанавливаем режим полной буферизации, размер буфера
11     setvbuf( ptrFile, NULL, _IOFBF, 1024 );
12
13     //операции для работы с файлом
14
15     fclose (ptrFile);
16
17     return 0;
18 }
```


## Пример работы программы

В этом примере создаётся файл `file.txt`. Который связан с буферизованным потоком. Размер буфера 1024 байт, режим буферизации – полный. Поэтому, выходные данные в потоке должны записываться на физическое устройство, после того, как будет заполнен буфер.

 Обсудить на форуме (/topics/)

 Автор: admin (/forums/users/admin/)

 Дата: 10.09.2012

 Поделиться:

### Похожие статьи:

1. Функция `setbuf` (<http://cppstudio.com/post/1262/>)
2. Работа с файлами в C++ (<http://cppstudio.com/post/446/>)
3. Функция `fread` (<http://cppstudio.com/post/1641/>)
4. Тип данных `FILE` (<http://cppstudio.com/post/1544/>)
5. Обработка массива данных предметной области «штатное расписание» (<http://cppstudio.com/post/1500/>)








## Комментарии

### Оставить комментарий

Вы должны войти (<http://cppstudio.com/wp-login.php?>

`redirect_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F1265%2F`), чтобы оставить комментарий.

## Translation

 (/post/1265/)Русский (/post/1265/)  
 (/uk/post/1265/)Українська (/uk/post/1265/)  
 (/en/post/1265/)English (/en/post/1265/)  
 (/de/post/1265/)Deutsch (/de/post/1265/)  
 (/be/post/1265/)Беларуская (/be/post/1265/)  
 (/kk/post/1265/)Қазақ тілі (/kk/post/1265/)  
 (/uz/post/1265/)O'zbek tili (/uz/post/1265/)  
 (/tr/post/1265/)Türkçe (/tr/post/1265/)

## 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>)