


[⬆ \(http://cppstudio.com\)](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 setbuf

Функция setbuf

 Оценка: **4,00** (голосов: 1)

Чтобы проголосовать, вы должны зарегистрироваться.

Прототип функции setbuf:

```
1 void setbuf( FILE * filestream, char * buffer );
```

Заголовочный файл

Название	Язык
stdio.h	C
cstdio	C++

Описание

Устанавливает режим буферизации, который будет использоваться для операций ввода/вывода с заданным потоком. При полной буферизации потока, в качестве аргумента передается указатель на строку буферизации. Если же, аргумент параметра `buffer` - `NULL`, буферизация для потока отключается, и данный поток становится небуферизованным.

Эта функция должна быть вызвана для файла, связанного с потоком, где-то между операциями открытия и закрытия файла.

Параметр `buffer` должен указывать на символьный массив, по крайней мере, размером `BUFSIZ` байт (`BUFSIZ` это константа, определенная в `<cstdio>`).

С полной буферизацией потоков, операций записи не предназначены быть написано непосредственно на устройстве, связанные с ними, данные накапливаются в буфере и записываются в устройство в качестве блока, когда он наполнен. Это письмо из всего блока не может быть принужден к промывке поток, который можно сделать по телефону `fflush` или закрытии файла (`fclose`). Все буферы и покраснел, когда программа завершает работу.

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.

Чтобы изменить строку буферизации потока, используйте функцию `setvbuf`.

Параметры:

- **filestream**

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

- **buffer**

Выделенный пользователем буфер. Он должен иметь длину не менее `BUFSIZ` байт. `BUFSIZ` – макроконстанта, определенная в `<stdio>` специально разработана для указания длины массива. Кроме того, указатель `NULL` может быть задан для отключения буферизации.

Возвращаемое значение

нет

Пример: исходный код программы

```
1  //приме риспользования функции setbuf
2
3  #include <iostream>
4  #include <stdio>
5
6  int main ()
7  {
8      char buffer[BUFSIZ];
9      FILE * ptrFile1 = fopen("file1.txt", "w");
10     * ptrFile2 = fopen("file2.txt", "a");
11
12     setbuf( ptrFile1, buffer ); // привязываем буфер
13     fputs("Эта строка отправлена в файловый буфер", ptrFile1);
14     fflush (ptrFile1); // очистить буфер
15
16     setbuf ( ptrFile2, NULL ); // отключить буфериз
17     fputs ("Эта строка отправлена в небуферизованный поток", ptrFile2);
18
19     fclose (ptrFile1);
20     fclose (ptrFile2);
21
22     return 0;
23 }
```

Пример работы программы

Открываются два файла с доступом записи. Поток, связанный с файлом `file1.txt` буферизован, выделенной символьной строкой `buffer`.

Второй буфер, связанный с файлом `file2.txt`, установлен как небуферизованный, поэтому вывод записывается на устройство как можно скорее.

Конечный результат, в обоих случаях одинаков, как с буфером так и без буферизации потока.

Обсудить на форуме (/topics/)

Автор: admin (/forums/users/admin/)

Дата: 10.09.2012

Поделиться:

Похожие статьи:

1. Функция setvbuf (<http://cppstudio.com/post/1265/>)
2. Работа с файлами в C++ (<http://cppstudio.com/post/446/>)
3. Функция fread (<http://cppstudio.com/post/1641/>)
4. Функция ungetc (<http://cppstudio.com/post/1644/>)
5. Функция rewind (<http://cppstudio.com/post/1576/>)









Комментарии

Оставить комментарий

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

[redirect_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F1262%2F](http://cppstudio.com/wp-login.php?redirect_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F1262%2F)), чтобы оставить комментарий.

Translation

 (/post/1262/)Русский (/post/1262/)
 (/uk/post/1262/)Українська (/uk/post/1262/)
 (/en/post/1262/)English (/en/post/1262/)
 (/de/post/1262/)Deutsch (/de/post/1262/)
 (/be/post/1262/)Беларуская (/be/post/1262/)
 (/kk/post/1262/)Қазақ тілі (/kk/post/1262/)
 (/uz/post/1262/)O'zbek tili (/uz/post/1262/)
 (/tr/post/1262/)Türkçe (/tr/post/1262/)

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>)