

Search:

Go

C++
Информация
Руководства
Ссылка
Статьи
Форум

Ссылка
Библиотека C:
<cassert> (assert.h)
<cctype> (ctype.h)
<cerrno> (errno.h)
<cfenv> (fenv.h)
<cfloat> (float.h)
< cinttypes> (inttypes.h)
<ciso646> (iso646.h)
<climits> (limits.h)
<locale> (locale.h)
<cmath> (math.h)
< csetjmp> (setjmp.h)
<csignal> (signal.h)
< cstdarg> (stdarg.h)
< cstdbool> (stdbool.h)
< cstddef> (stddef.h)
< cstdint> (stdint.h)
< cstdio> (stdio.h)
< cstdlib> (stdlib.h)
< cstring> (string.h)
< ctmath> (tmath.h)
<ctime> (time.h)
< cuchar> (uchar.h)
< cwchar> (wchar.h)
< cwstring> (wstring.h)
Контейнеры:
Ввод/вывод:
Многопоточность:
Другое:

<stdio> (stdio.h)
функции:
clearerr
fclose
feof
ффеппор
fflush
fgetc
fgetpos
fgets
fopen
fprintf
fputc
fputs
fread
freopen
fscanf
fseek
fsetpos
ftell
fwrite
getc
getchar
получает
perror
printf
putc
putchar
ставит
удалить
переименовать
перемотка назад
scanf
setbuf
setvbuf
snprintf
sprintf
sscanf
tmpfile
tmpnam
ungetc
vfprintf
vfscanf
vprintf
vscanf
vsprintf
vsprintf

функция

fflush

<stdio>

int fflush (ФАЙЛ * поток);

Поток промывки

Если данный *поток* был открыт для записи (или если он был открыт для обновления, а последняя операция ввода-вывода была операцией вывода), любые неписанные данные в его выходном буфере записываются в файл.

Если *stream* является нулевым указателем, все такие потоки сбрасываются.

Во всех остальных случаях поведение зависит от конкретной реализации библиотеки. В некоторых реализациях очистка потока, открытого для чтения, приводит к очистке его входного буфера (но это не переносимое ожидаемое поведение).

Поток остается открытым после этого вызова.

Когда файл закрыт, либо из-за вызова `fclose` или, поскольку программа завершается, все буферы, связанные с ней, автоматически очищаются.

Параметры

поток

Указатель на *файловый* объект, указывающий буферизованный поток.

Возвращаемое значение

Нулевое значение указывает на успех.

При возникновении ошибки *возвращается EOF* и устанавливается индикатор ошибки (см. `feof`).

Пример

В файлах, открытых для обновления (т. е. Открытых как для чтения, так и для записи), поток должен быть очищен после операции вывода перед выполнением операции ввода. Это может быть сделано либо путем перестановки (`fseek`, `fsetpos`, `rewind`), либо путем явного вызова `fflush`, как в этом примере:

```
1 /* fflush example */
2 #include <stdio.h>
3 char mybuffer[80];
4 int main()
5 {
6     FILE * pFile;
7     pFile = fopen ("example.txt","r+");
8     if (pFile == NULL) perror ("Error opening file");
9     else {
10         fputs ("test",pFile);
11         fflush (pFile); // flushing or repositioning required
12         fgets (mybuffer,80,pFile);
13         puts (mybuffer);
14         fclose (pFile);
15         return 0;
16     }
17 }
```

Редактировать и запускать

См. Также

fclose	Закрыть файл (функция)
fopen	Открыть файл (функция)
setbuf	Установить буфер потока (функция)
setvbuf	Изменение буферизации потока (функция)

vsscanf

objects:

stderr

stdin

stdout

types:

FILE

fpos_t

size_t

macro constants:

BUFSIZ

EOF

FILENAME_MAX

FOPEN_MAX

L_tmpnam

NULL

TMP_MAX

[Home page](#) | [Privacy policy](#)

© cplusplus.com, 2000-2021 - All rights reserved - v3.2
[Spotted an error? contact us](#)