

Поиск:

Вперед

Зарегистрирован как: legioner9

Учетная запись  
Выход

Ссылка <stdio> feof

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> (tgmath.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 ferror fflush fgetc fgetpos fgets fopen fprintf fputc fputs fread freopen fscanf fseek fsetpos ftell fwrite getc getchar получает perror printf putc putcchar ставит удалить переименовать перемотка scanf setbuf setvbuf snprintf sprintf sscanf tmpfile tmpnam ungetc vfprintf vfscanf vprintf vscanf vsnprintf vsprintf

функция

feof

<stdio>

`int feof ( ФАЙЛ * поток );`  
**Проверить индикатор конца файла**  
Проверяет, установлен ли индикатор *конца файла*, связанный с *поток*ом, и возвращает значение, отличное от нуля, если оно установлено.  
  
Этот индикатор обычно устанавливается предыдущей операцией в *потоке*, которая пыталась прочитать в конце файла или после него.  
  
Обратите внимание, что внутренний индикатор положения потока может указывать на *конец файла* для следующей операции, но все же индикатор *конца файла* может не устанавливаться до тех пор, пока операция не попытается прочитать в этой точке.  
  
Этот индикатор очищается вызовом `clearerr`, `rewind`, `fseek`, `fsetpos` или `freopen`. Хотя если индикатор *положения* если такой вызов не будет изменен, то следующая операция ввода-вывода, скорее всего, снова установит индикатор.

Параметры

поток  
Указатель на *файловый* объект, идентифицирующий поток.

Возвращаемое значение

Ненулевое значение возвращается в том случае, *если установлен индикатор конца файла*, связанный с *поток*ом, в противном случае возвращается ноль.

Пример

```
1 /* feof example: byte counter */
2 #include <stdio.h>
3
4 int main ()
5 {
6     FILE * pFile;
7     int n = 0;
8     pFile = fopen ("myfile.txt","rb");
9     if (pFile==NULL) perror ("Error opening file");
10    else
11    {
12        while (fgetc(pFile) != EOF) {
13            ++n;
14        }
15        if (feof(pFile)) {
16            puts ("End-of-File reached.");
17            printf ("Total number of bytes read: %d\n", n);
18        }
19        else puts ("End-of-File was not reached.");
20        fclose (pFile);
21    }
22    return 0;
23 }
```

Редактировать  
и  
запустить

Этот код открывает файл под названием `myfile.txt`, и подсчитывает количество символов, которые он содержит, читая их все один за другим. Программа проверяет, был ли достигнут конец файла, и если да, то выводит общее количество прочитанных байтов.

См. Также

<code>clearerr</code>	Очистить индикаторы ошибок ( <a href="#">функция</a> )
<code>ferror</code>	Проверить индикатор ошибки ( <a href="#">функция</a> )

vsscanf

**Объекты:**

stderr

stdin

stdout

**типы:**

ФАЙЛ

fpos\_t

size\_t

**macro constants:**

BUFSIZ

EOF

FILENAME\_MAX

OPEN\_MAX

L\_tmpnam

NULL

TMP\_MAX

---

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

© cplusplus.com, 2000-2021 - All rights reserved - v3.2

[Spotted an error? contact us](#)