

C++

Информация
Руководства
Ссылка
Статьи
Форум

Библиотека C:
 <assert> (assert.h)
 <ctype> (ctype.h)
 <errno> (errno.h)
 <fcntl> (fcntl.h)
 <float> (float.h)
 <inttypes> (inttypes.h)
 <iso646> (iso646.h)
 <limits> (limits.h)
 <locale> (locale.h)
 <math> (math.h)
 <setjmp> (setjmp.h)
 <signal> (signal.h)
 <stdarg> (stdarg.h)
 <stdbool> (stdbool.h)
 <stddef> (stddef.h)
 <stdint> (stdint.h)
 <stdio> (stdio.h)
 <stdlib> (stdlib.h)
 <string> (string.h)
 <tgmath> (tgmath.h)
 <time> (time.h)
 <uchar> (uchar.h)
 <wchar> (wchar.h)
 <wctype> (wctype.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
putchar
ставит
удалить
переименовать
перемотка назад
scanf
setbuf
setvbuf
snprintf
sprintf
sscanf
tmpfile
tmpnam
ungetc
vfprintf
vfprintf
vprintf
vscanf
vsnprintf
vsprintf
vsscanf
Объекты:
stderr
stdin
stdout
типъ:
ФАЙЛ
fpos_t

функция **fopen** <stdio>

```
FILE * fopen ( const char * filename, const char * mode );
```

Открыть файл

Открывает файл, имя которого указано в параметре *filename*, и связывает его с потоком, который может быть идентифицирован в будущих операциях по возвращаемому указателю ФАЙЛА.

Разрешенные операции в потоке и способ их выполнения определяются параметром *mode*.

Возвращаемый поток по умолчанию полностью буферизуется, если известно, что он не ссылается на интерактивное устройство (см. `setbuf`).

Возвращаемый указатель может быть отделен от файла с помощью вызова `fclose` или `freopen`. Все открытые файлы автоматически закрываются при обычном завершении программы.

Рабочая среда поддерживает по крайней мере Файлы FOPEN_MAX открываются одновременно.

Параметры

имя файла

Строка C, содержащая имя файла, который нужно открыть.

Его значение должно соответствовать спецификациям имени файла рабочей среды и может включать путь (если поддерживается системой).

режим

Строка C, содержащая режим доступа к файлу. Это может быть:

"r"	читать: Открыть файл для операций ввода. Файл должен существовать.
"w"	запись: Создание пустого файла для операций вывода. Его содержимое отбрасывается, и файл рассматривается как новый.
"a"	добавить: Открыть файл для вывода в конце файла. Операция расширяя его. Операции репозиционирования (fseek, fsetpos) не работают.
"r+"	чтение/обновление: Откройте файл для обновления (как чтение и запись). Файл должен существовать.
"w+"	запись/обновление: создайте пустой файл и откройте его для чтения и записи. Если файл с таким же именем уже существует, его содержимое отбрасывается, и создается новый пустой файл.
"a+"	добавление/обновление: Откройте файл для обновления (как для ввода, так и для вывода). Все операции ввода, записывающими данные в конце файла. Операции перемещения (fseek, fsetpos, rewind) влияют на следующие операции ввода, но операции вывода перемещают позицию обратно в конец файла. Файл создается, если он не существует.

С помощью спецификаторов режима выше файл открывается как *текстовый файл*. Чтобы открыть файл в виде *двоичного файла*, в строку mode должен быть включен символ "b". Этот дополнительный символ "b" может быть либо добавлен в конце строки (таким образом, делая следующие составные режимы: "rb", "wb", "ab", "r+b", "w+b", "a+b"), либо вставлен между буквой и знаком "+" для смешанных режимов ("rb+", "wb+", "ab+").

Новый стандарт C (C2011, который не является частью C++) добавляет новый стандартный подспецификатор ("x"), который может быть добавлен к любому спецификатору "w" (для формирования "wx", "wbx", "w+x" или "w+bx"/"wb+x"). Этот подспецификатор заставляет функцию терпеть неудачу, если файл существует, вместо того, чтобы перезаписывать его.

Если за последовательностью следуют дополнительные символы, поведение зависит от реализации библиотеки: некоторые реализации могут игнорировать дополнительные символы, так что, например, дополнительный "t" (иногда используемый для явного указания *текстового файла*) принят.

В некоторых реализациях библиотек открытие или создание текстового файла с режимом обновления может обрабатывать поток как двоичный файл.

Текстовые файлы – это файлы, содержащие последовательности строк текста. В зависимости от среды, в которой работает приложение, некоторые специальные преобразования символов могут происходить в операциях ввода/вывода в *текстовом режиме*, чтобы адаптировать их к системному формату текстового файла. Хотя в некоторых средах преобразования не происходят и *текстовые файлы* и *двоичные файлы* обрабатываются одинаково, использование соответствующего режима улучшает переносимость.

Для файлов, открытых для обновления (тех, которые содержат знак "+"), для которых разрешены операции ввода и вывода, поток должен быть сброшен (`flush`) или перемещен (`fseek`, `fsetpos`, `rewind`) перед операцией чтения, которая следует за операцией записи. Поток должен быть перемещен (`fseek`, `fsetpos`, `rewind`) перед операцией записи, которая следует за операцией чтения (всякий раз, когда эта операция не достигла конца файла).

Возвращаемое значение

Если файл успешно открыт, функция возвращает указатель на **файловый** объект, который можно использовать для идентификации потока при будущих операциях.

В противном случае возвращается нулевой указатель.

В большинстве реализаций библиотек переменная `errno` также устанавливается в системный код ошибки при сбое.

Пример

```
1 /* fopen example */
2 #include <stdio.h>
3 int main ()
4 {
5     FILE * pFile;
6     pFile = fopen ("myfile.txt", "w");
7     if (pFile!=NULL)
8     {
9         fputs ("fopen example",pFile);
10        fclose (pFile);
11    }
```

Редактирование
и
запуск

макроконстанты:
BUFSIZ
EOF
FILENAME_MAX
FOPEN_MAX
L_tmpnam
NULL
TMP_MAX

```
12 | return 0;  
13 }
```

См. Также

fclose	Заккрыть файл (функцию)
freopen	Повторное открытие потока с другим файлом или режимом (функцией)
setbuf	Установить буфер потока (функция)
setvbuf	Изменить буферизацию потока (функцию)
tmpfile	Открыть временный файл (функция)
tmpnam	Создание временного имени файла (функции)