



- [Главная](#)
- [История](#) ▾
  - [Домеханический этап](#) ▸
    - [Зарождение счета](#)
    - [Бирки](#)
    - [Китайские счетные палочки](#)
    - [Узелковая письменность](#)
    - [Абак](#)
    - [Китайская счетная доска](#)
    - [Суаньпань](#)
    - [Соробан](#)
    - [Счет на линиях](#)
    - [Счет костями](#)
    - [Дощаный счет](#)
    - [Счеты](#)
    - [Палочки Непера](#)
    - [Логарифмы](#)
    - [Логарифмическая линейка](#)
  - [Механический этап](#) ▸
    - [Счетная машина Леонардо](#)
    - [Вычисляющие часы](#)
    - [Паскалин](#)
    - [Счетная машина Морленда](#)
    - [Калькулятор Лейбница](#)
    - [Рабдологический абак](#)
    - [Арифмометр Полени](#)
    - [Машина Перейры](#)
    - [Счетная машина Якобсона](#)
  - [Первое поколение ЭВМ](#)
  - [Второе поколение ЭВМ](#)
  - [Третье поколение ЭВМ](#)
  - [Персональные ЭВМ](#)
- [Устройство ПК](#) ▾
  - [Материнская плата](#)
  - [Процессор](#) ▸
    - [Устройство процессора](#)
    - [Хронология ЦП Intel](#)
  - [Оперативная память](#) ▸
    - [Типы ОЗУ](#)
    - [Динамическая оперативная память](#)
    - [Статическая оперативная память](#)
    - [Магниторезистивная оперативная память](#)
  - [Постоянная память](#)
  - [Периферия](#)
- [ВС и Сети](#) ▾
  - [Надежность ВС](#) ▸
    - [Надежность](#)
    - [Повышение надежности](#)
    - [Резервирование оборудования](#)
    - [Контроль исправности](#)
    - [Высоконадежные ВС](#)
    - [Программное резервирование](#)
    - [Избыточное кодирование](#)
    - [Контрольная сумма CRC](#)
    - [Коды Хемминга](#)
  - [Бортовые ВС](#)
  - [Сети](#)
- [Разработка ПО](#) ▾
  - [Системы контроля версий](#) ▸
    - [Введение в СКВ](#)
    - [Обзор СКВ](#)
    - [Начинаем работать с СКВ GIT](#)
  - [Виртуальные ЭВМ](#) ▸
    - [Виртуальные машины](#)

[Поиск](#)[Содержание](#)

- [Технология виртуализации](#)
- [Начинаем работать с VMware](#)
- [Начинаем работать с VirtualBox](#)
- [Язык Си](#)
- [Стиль Си](#)
- [Полезное ▾](#)
  - [Бесплатное ПО первой необходимости](#)
  - [Заставки на рабочий стол ▸](#)
    - [Абстракция](#)
    - [Архитектура](#)
    - [Природа](#)

[Главная](#) ▸ [Разработка ПО](#) ▸ [функции языка Си](#)

## Описание функций языка Си

" F " [A](#)[B](#)[C](#)[D](#)[E](#)[F](#)[G](#)[H](#)[I](#)[J](#)[K](#)[L](#)[M](#)[N](#)[O](#)[P](#)[Q](#)[R](#)[S](#)[T](#)[U](#)[V](#)[W](#)[X](#)[Y](#)[Z](#)

[fabs](#), [fabsf](#), [fabsl](#) ▸

**fseek** – установка позиции в потоке данных.

[fclose](#) ▸

**Синтаксис:**

[fcloseall](#) ▸

```
#include <stdio.h>
```

```
int fseek(FILE *stream, long int offset, int whence);
```

[fdim](#), [fdimf](#), [fdiml](#) ▸

**Аргументы:**

[fdopen](#) ▸

stream – указатель на управляющую таблицу потока данных.

[fgetround](#) ▸

offset – смещение позиции.

whence – точка отсчета смещения.

[feof](#) ▸

**Возвращаемое значение:**

[ferror](#) ▸

0 – при успешной установке позиции.

[fesetround](#) ▸

Отличное от нуля значение, если при работе функции произошли ошибки. При этом переменной errno будет присвоен код ошибки:

[fgetc](#) ▸

[EINVAL] – неверное значение аргумента whence

[fgets](#) ▸

[ESPIPE] – недопустимое значение аргумента offset

[fileno](#) ▸

**Описание:**

Функция fseek() устанавливает позицию в потоке данных, заданным аргументом stream. Относительно установленной позиции будет осуществляться чтение и запись данных.

[finite](#), [finitel](#), [finitel](#) ▸

[fflush](#) ▸

Точка отсчета устанавливаемой позиции определяется аргументом whence, который может принимать значения:

[floor](#), [floorf](#), [floorl](#) ▸

SEEK\_SET – смещение отсчитывается от начала файла

SEEK\_CUR – смещение отсчитывается от текущей позиции

[fma](#), [maf](#), [mal](#) ▸

SEEK\_END – смещение отсчитывается от конца файла

[fmax](#), [fmaxf](#), [fmaxl](#) ▸

Смещение задается аргументом offset, причем положительное значение аргумента означает смещения вправо от указанной аргументом whence позиции, отрицательное – смещение влево.

[fmin](#), [fminf](#), [fminl](#) ▸

[fmod](#), [fmodf](#), [fmodl](#) ▸

Для двоичных потоков данных, смещение (offset) – это количество байт.

[fopen](#) ▸

Для текстовых потоков данных смещение (offset) должно быть равно 0, либо получено с помощью функции ftell(), при этом точка отсчета (whence) должна иметь значение SEEK\_SET.

[fpclassify](#) ▸

**Пример:**

[fputc](#) ▸

В примере считывается строка из начала файла, а затем из позиции, смещенной относительно начала файла на 5 байт. Смещение задается с помощью функции fseek. В файле записана следующая строка: 123456789.

[fputs](#) ▸

[freopen](#) ▸

[frexp](#), [frexpf](#), [frexpl](#) ▸

[fseek](#) ▸

[fseeko](#) ▸

[ftell](#) ▸

[ftello](#) ▸

```
#include <stdio.h> //fopen, fclose, printf, fseek, fgets

int main (void)
{
    // Переменная, в которую будет сохранен указатель
    // на управляющую таблицу открываемого потока данных
    FILE *mf;
    //Массив в который будет записана считанная из потока данных строка
    char str[50];

    // Открытие файла
    printf ("Открытие файла: ");
    mf=fopen ("myfile/test.txt","r+");
    // Проверка открытия файла
    if (mf == NULL) printf ("ошибка\n");
```

```
else printf ("выполнено\n");

//Установка текущей позиции
printf ("    Установка позиции на начало файла: ");
if (fseek (mf,0,SEEK_SET)==0)
    printf ("выполнено\n");
else
    printf ("ошибка\n");

//Чтение строки из файла
printf ("    Считываем строку: ");
if (fgets (str, sizeof (str), mf)==NULL)
    printf ("строка не считана\n");
else
    printf ("%s\n",str);

//Установка текущей позиции
printf ("    Установка позиции на пятый байт: ");
if (fseek (mf,5,SEEK_SET)==0)
    printf ("выполнено\n");
else
    printf ("ошибка\n");

//Чтение строки из файла
printf ("    Считываем строку: ");
if (fgets (str, sizeof (str), mf)==NULL)
    printf ("строка не считана\n");
else
    printf ("%s\n",str);

// Закрытие файла
printf ("Закрытие файла: ");
if ( fclose (mf) == EOF) printf ("ошибка\n");
else printf ("выполнено\n");

return 0;
}
```

**Результат:**

Вывод на консоль:

Открытие файла: выполнено  
Установка позиции на начало файла: выполнено  
Считываем строку: 123456789  
Установка позиции на пятый байт: выполнено  
Считываем строку: 6789  
Закрытие файла: выполнено

**Смотри так же:**

[fseek](#)  
[fgetpos](#)  
[fseeko](#)  
[fsetpos](#)  
[ftell](#)  
[ftello](#)  
[rewind](#)

