

# File Streams

# Хранение информации

Для программиста работа с файлами имеет очень большое значение. Длительное хранение информации только в оперативной памяти невозможно. Файл же хранит информацию на диске / SD-карте, что позволяет обратиться к ней в любой момент.

# Определение файла

**Файл - это именованная область данных на носителе информации (конечная совокупность байт).**

- Файл не может располагаться на диске / карте непрерывно, однако пользователю файл предоставляется цельным блоком последовательной байтовой информации
- Название файла не может содержать символы < > : " / \ |
- Большинство файлов обладает расширением - сочетанием символов, с помощью которых операционная система определяет тип файла
- У каждого файла есть так называемые атрибуты (например - скрытый, системный, архивный, является папкой, только для чтения и тд).

# Сущности, связанные с файлами

- **Дескриптор файла** - при открытии файла операционная система возвращает уникальный номер, с помощью которого выполняются все файловые операции. По их завершении файл закрывается, и дескриптор теряет смысл.
- **Файловый указатель** - число, являющееся смещением относительно нулевого байта в файле. Обычно по этому адресу осуществляется чтение/запись. При выполнении операций чтения/записи файловый указатель смещается на число прочитанных (записанных) байт. Последовательный вызов операций чтения таким образом позволяет прочитать весь файл, не заботясь о его размере.
- **Файловый буфер** - операционная система осуществляет кэширование файловых операций в специальном буфере (участке памяти). При закрытии файла буфер высвобождается.

# Виды файлов

- **Бинарный файл** - это самый обычный файл, просто совокупность байтов. Условно можно рассматривать такой файл как массив.
- **Текстовый файл** - это файл, который при интерпретации значений его байтов, как кодов его символов и представлении его на экране в виде совокупности этих символов - образует осмысленный текст. По сути, это совокупность строк. Разделителем строк является символ перевода строки.

# Сравнительная таблица

<b>Текстовый файл</b>	<b>Двоичный файл</b>
Число, записанное в файл - запишется как текст, то есть размер занятого пространства будет равен количеству цифр в этом числе.	Число, записанное в файл - запишется в двоичном формате, и размер занятого пространства будет равен размеру типа данных в этом числе.
Прочитать данный файл можно с помощью любого текстового редактора.	Прочитать файл можно только с помощью особой программы.
При потере нескольких байт, информацию можно восстановить по смыслу.	Потеря нескольких байт в двоичном файле может быть необратима.
Считывание информации программным путем затруднено, так как файл представляет собой единый текстовый блок.	Считывание информации программным путем облегчено, так как файл представляет собой набор блоков информации с конкретными типами данных.
Используется как файл для чтения, редактирования, вывода на печать.	Используется как файл для удобного чтения, хранения, записи информации программным путём.

# FILE\*

Базовый адрес блока памяти файла можно сохранить, используя указатель типа FILE\*. Это специальная переменная, которая хранит текущую позицию в файле. Изначально эта позиция равна 0, но потом при чтении и записи она сдвигается. При желании эту позицию можно будет вернуть в начало файла, или же в любое другое место. Файловый указатель можно представить как самый обыкновенный курсор | в любом текстовом редакторе, который позволяет понять, в каком месте файла мы сейчас находимся.

# Режимы работы с файлом

Для создания или открытия уже созданного файла используется функция `fopen`:

**FILE\* f = fopen("путь к файлу", "режим открытия");**

Путь к файлу может быть относительным или абсолютным адресом.

Пример абсолютного адреса - `C:\Users\Саша\Desktop\for save\!`

`Работа\Материалы\C\Практика\Задания по группам\ЕКО1691.txt`.

Пример относительного адреса – `ЕКО1691.txt`, но при этом документ должен находиться в непосредственной близости от `exe`-файла приложения.

Режимы:

**r** - чтение, если файла нет, то данная функция генерирует ошибку (возвращает 0) - сразу программа не вылетит, но она поломается потом, когда вы попытаетесь что-то считать из файла или записать в него что-нибудь

**w** - запись, если файла нет, то файл создаётся, если есть исходное содержимое удаляется

**a** - добавление в конец, если файла нет, то он создаётся

**r+** - чтение и запись (файл должен существовать!)

**w+** - чтение и запись (принцип работы как у `w`)

**a+** - добавление и чтение (принцип работы как у `a`)

# Открытие файла

<https://git.io/vACAZ>

# Чтение по одному символу

<https://git.io/vACAM>

# Запись по одному символу

<https://git.io/vACAd>

# Работа построчно

<https://git.io/vACxT>

# Перемещение по файлу

<https://git.io/vACxc>

# Считывание файла целиком

<https://git.io/vACx7>

# Создание 100500 файлов

<https://git.io/vACjA>

`fwrite` + двоичные файлы

<https://git.io/vAWve>

# Запись 2D-массива в файл

<https://git.io/vAWvq>

# Создание папок

<https://git.io/vAWv8>

# Поиск файлов

<https://git.io/vAWv1>

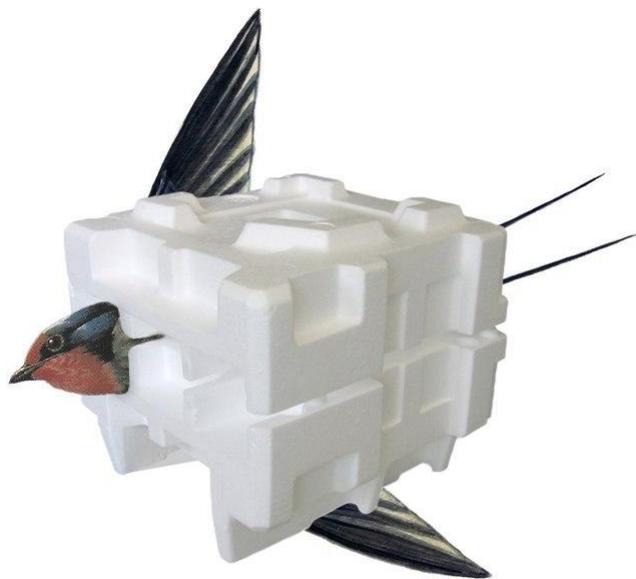
# Практика

- Игра «Составь предложение». Есть файлы: Кто, Где, Когда, С кем, Что делали, Что получилось, Мораль. Пользуясь `rand()`, программа составляет шуточное предложение и показывает его на экран.
- В файле в столбик хранятся русские существительные. Пользователь вводит слово. Задача программы подобрать из файла рифму к указанному слову.

# Практика

Написать приложение, которое будет составлять новые смешные слова (без рисунков!) вроде «пенопласточки» 😊

Словарь: <https://git.io/vXqa0>



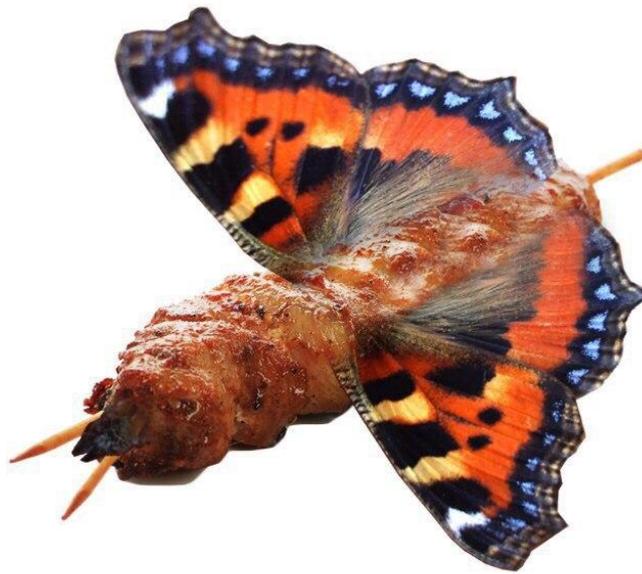
**ПЕНОПЛАСТОЧКА**



**АНТИЛОПАСТЬ**



**ПАУКСУС**



**ЛЮЛЯ-КЕБАБОЧКА**



**ЛАМАРЕТТО**



**ОПОССУМКА**

автомобильярд  
автотранспортал  
агентствол  
азбукашка  
акварельс  
антиквартирант  
аптекарьера  
аристократкость  
аттестатуя  
беднотариус  
бешенствол  
боеприпасынок  
бутербродяга  
верблюдце  
вертикальян  
виолончелистопад  
водительняшка  
гимнастроение  
гитаракан  
говядинамик  
горизонттик  
госпожарник  
заботаника  
завтракторист  
иммигрантье  
исповедьма  
калекарь  
карательняшка  
кикимораль  
кобельё  
комфортетьяно  
макакао  
метеоретик  
методикарь  
мотоциклон

# Практика

- Создать класс СПРАВОЧНИК со следующими полями: Название фирмы, Владелец, Телефон, Адрес, Род деятельности. Реализовать следующие возможности: Поиск по названию, Поиск по владельцу, Поиск по номеру телефона, Поиск по роду деятельности, Показ всех записей и добавление. Вся информация, естественно, хранится в файле с возможностью дозаписи.

# Практика

- Написать игру "Поле чудес"
- Написать игру "Виселица"
- Написать игру "Балда"