

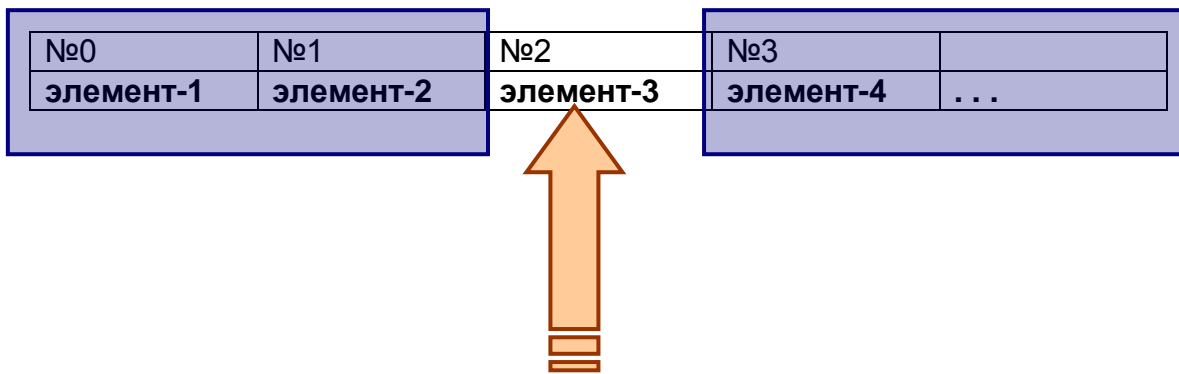
## Файлы

Файл – это область памяти на внешнем носителе, в котором хранится некоторая информация. В него можно поместить данные или извлечь их оттуда.

### Способ доступа

Файлы - бесконечный список значений одного и того же (базового) типа. Все элементы файла пронумерованы, начальный элемент имеет нулевой номер.

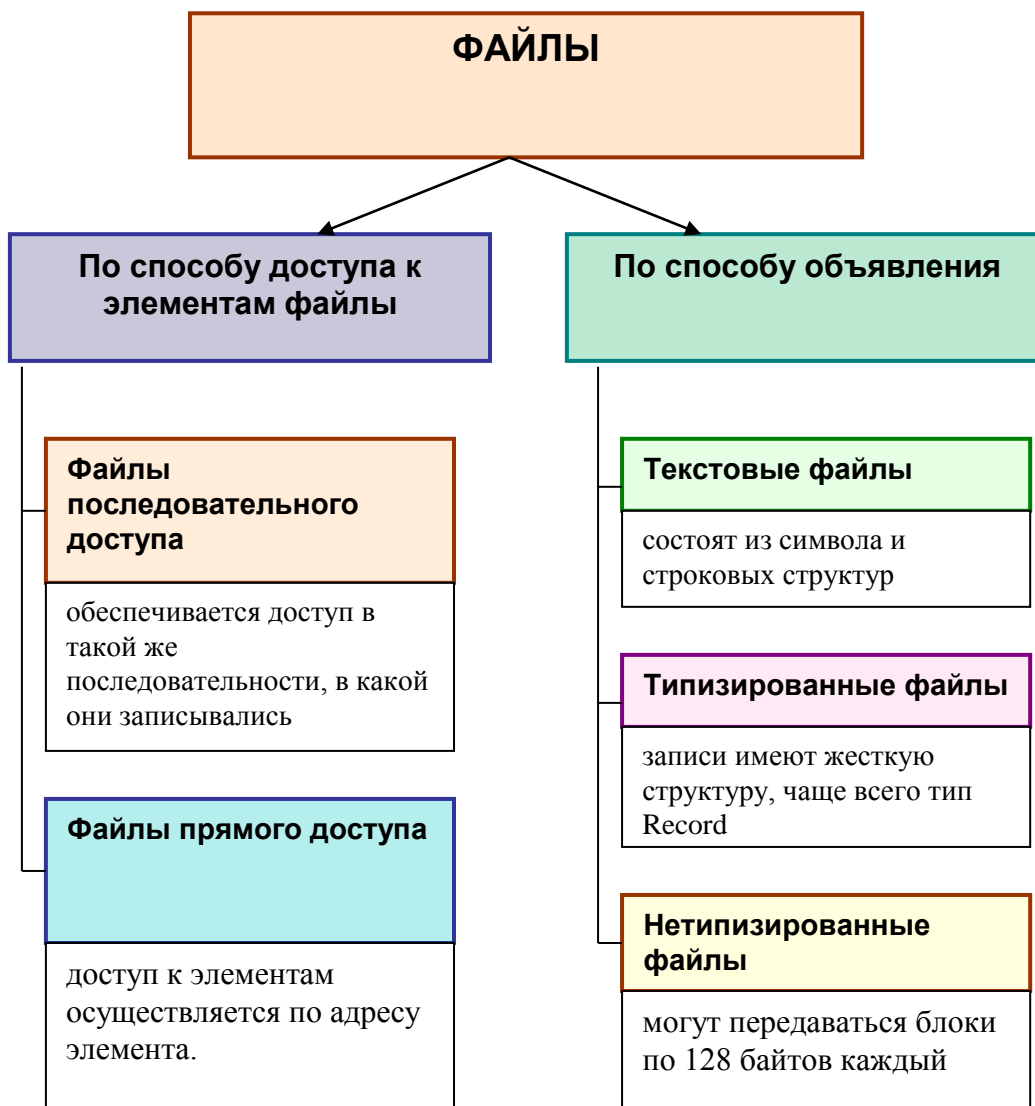
### Файл



Текущий указатель

В любой момент времени доступен только один элемент файла, на который ссылается **текущий указатель**.

Чтение из файла, запись в файл производятся поэлементно. В результате операций текущий указатель может перемещаться.



### Особенности

	<b>Последовательный доступ</b>
Поиск нужного элемента	Начиная с нулевого элемента, указатель обработки перемещается до тех пор, пока не будет указывать на искомый элемент.
	Нельзя <u>одновременно читать</u> данные из файла и <u>записывать</u> данные в файл, т.к. при чтении указатель помещен на данный элемент, а при записи находится в конце файла.

## Текстовые файлы

Структура текстовых файлов рассматривается как последовательность символов, разбитая на строки длиной от 0 до 256 символов.

Каждая строка завершается **маркером конца строки** – последовательность двух символов:

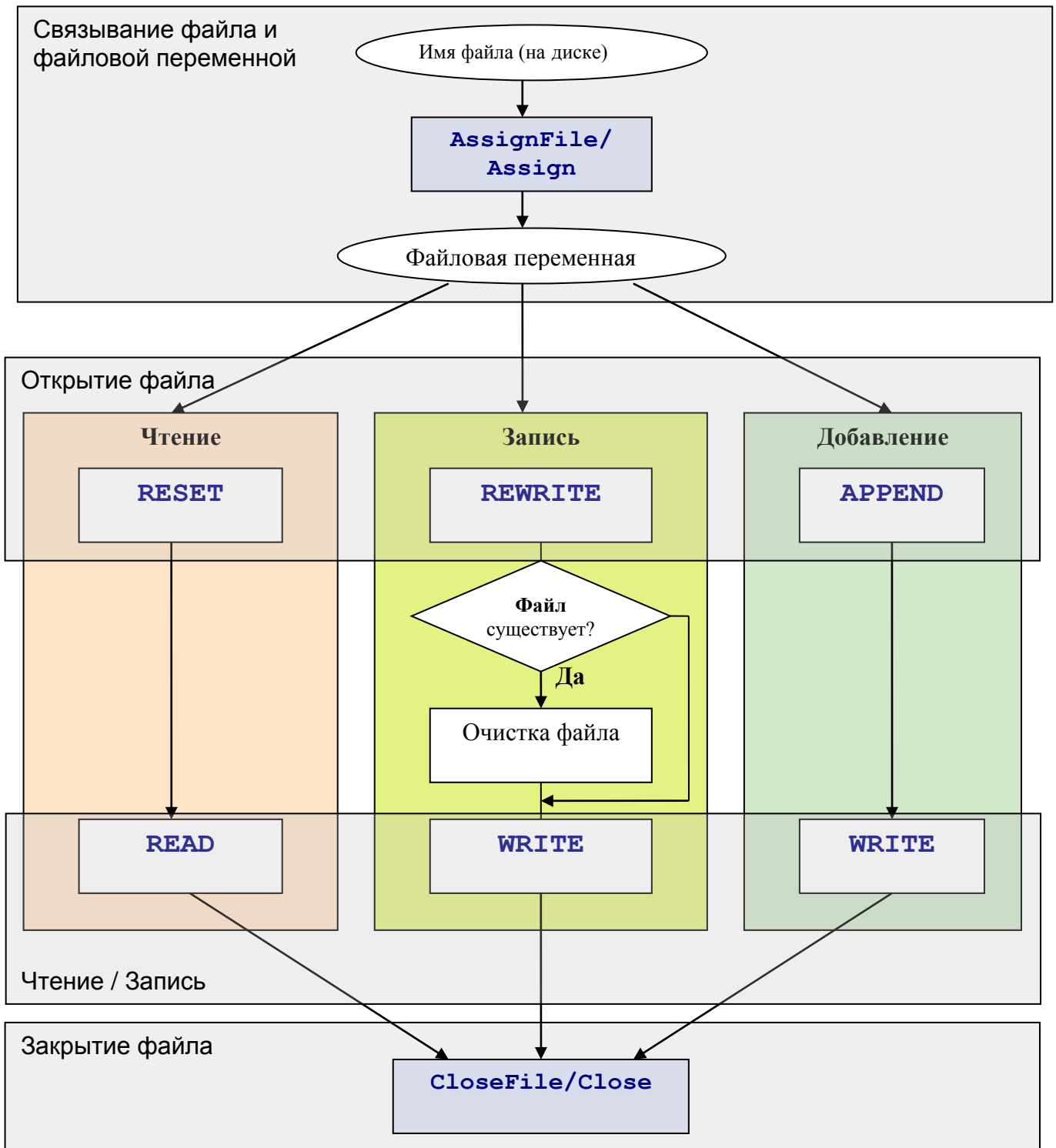
- перевод строки CHR(13),
- возврат каретки CHR(10).

Для описания используется стандартный тип:

Компилятор	Delphi, Lazarus	FreePascal, PascalABC
Стандартный файловый тип	<b>TextFile</b>	<b>Text</b>
Пример	<b>Var On_File: TextFile;</b>	<b>Var On_File: Text;</b>
Связывание файлов	<b>AssignFile</b>	<b>Assign</b>
Закрытие файлов	<b>CloseFile</b>	<b>Close</b>

## Особенности текстовых файлов

- Работа с текстовыми файлами организуется построчно.
- При чтении и записи числа преобразуются автоматически.
- Чтение и запись являются строго последовательными.
- Нельзя **одновременно читать** данные из файла и **записывать** данные в файл, т.к. при чтении указатель помещен на данный элемент, а при записи находится в конце файла.



**Основные понятия**

<b>Текстовый тип</b>		
	<b>TextFile/Text</b>	
<b>Связывание файлов</b>		
	<b>AssignFile</b> (<имя файловой переменной>, <имя файла>) <b>Assign</b> (<имя файловой переменной>, <имя файла>)	Устанавливает связь между переменной файлового типа (имя файла в программе Turbo Pascal) и именем физического файла, присваиваемого операционной системой. Имя файла может включать в себя и указание пути доступа к файлу. Имя физического файла должно быть конкретным и уникальным. Нельзя вставлять символы шаблонов '*' и '?'. Имя физического файла должно быть конкретным и уникальным. Нельзя вставлять символы шаблонов '*' и '?'.
<b>Открытие файла</b>		
Для чтения	<b>Reset</b> (<имя файловой переменной>)	Открытие файла уже существующего файла (если его нет, то возникнет сообщение об ошибке) для чтения. Процедуры открытия могут применяться многократно к одному и тому же файлу. Повторный вызов <b>Reset</b> переустановит последовательность чтения вновь на самый первый элемент файла.
Для записи	<b>Rewrite</b> (<имя файловой переменной>)	Открытие файла, который может не существовать. В этом случае процедура создает заданный файл для <u>записи</u> . Если же файл существует, происходит очищение его и файл открывается для последовательной записи в него.
<b>Закрытие файла</b>		
	<b>CloseFile</b> (<имя файловой переменной>) <b>Close</b> (<имя файловой переменной>)	Процедура закрывает открытый до этого логический файл, связь между файловой переменной и физическим файлом не изменится.
<b>Чтение из файла</b>		
	<b>Read</b> (<имя файловой переменной>, <переменные>) <b>ReadLn</b> (<имя файловой переменной>, <текстовые переменные>)	<текстовые переменные> - в которые будут помещаться читаемые из файла значения. <b>Read</b> - считывает очередной элемент строки. <b>ReadLn</b> - считывает очередной элемент строки и переходит на следующую строку. Если необходимо просто перейти к следующей строке, то используется процедура <b>ReadLn</b> (<имя файловой переменной>) Она переносит текущий указатель на первый элемент следующей строки.
<b>Запись в файл</b>		
	<b>Write</b> (<имя файловой переменной>, <переменные>) <b>WriteLn</b> (<имя файловой переменной>, <текстовые переменные>)	<текстовые переменные> - значения, которых будут помещаться в файл. <b>Write</b> – записывает элемент в текущую строку. <b>WriteLn</b> – записывает элемент в текущую строку и переходит к следующей строке. Если требуется перейти для записи на новую строку, то применяется процедура <b>WriteLn</b> (<имя файловой переменной>) Она ставит в конце данной строки признак ее конца и перемещает указатель на начало следующей строки. В разных строках может быть разное количество элементов.
<b>Организация контроля</b>		
Окончание строки	<b>EoLn</b> (<имя файловой переменной>)	Функция принимает значение <b>True</b> , если достигнут конец строки, и <b>False</b> в противном случае.
Окончание файла	<b>EoF</b> (<имя файловой переменной>)	Функция принимает значение <b>True</b> , если конец файла достигнут, и <b>False</b> в противном случае.

**Пример программы (FreePascal, PascalABC)****Var**

```
myFile : Text;  
text   : string;
```

**begin**

```
Assign (myFile, 'F:\MyDir\Test.txt');
```

```
ReWrite (myFile);           //Открыть файл для записи
```

```
WriteLn (myFile, 'Hello '); //Запись текста в файл
```

```
WriteLn (myFile, 444);
```

```
WriteLn (myFile, ' school');
```

```
Close (myFile);           //Закрытие файла
```

```
Reset (myFile);           //Открытие файла для чтения
```

```
While not Eof(myFile) do
```

```
Begin
```

```
  ReadLn (myFile, text);    //Показ содержимого файла
```

```
  WriteLn (text);
```

```
end;
```

```
Close (myFile);           //Закрытие файла раз
```

```
ReadLn;
```

```
end.
```

## Добавление записи

Используется для открытия файла с последующей записью в него новых строк в конец уже существующего файла.

**Append** (<имя файловой переменной>)

## Удаление файла

Стирает файл. Перед этой процедурой файл должен быть закрыт с помощью **CloseFile/Close**.

**Erase** (<имя файловой переменной>)

## Переименование файла

Файл переименовывается. В каталог вносится соответствующее изменение. Перед этой процедурой файл должен быть закрыт с помощью **CloseFile/Close**.

**Rename** (<имя файловой переменной>, <новое имя файла>)

**Пример (Delphi,Lazarus)**

Программа позволяет записать в файл некоторые данные, потом считать их из файла и дописать в него еще данные.

```
    Const N=10;
    Var myFile:TextFile;
        I,J:Integer;
Begin
    AssignFile(myFile,'Exl.dat');
    Rewrite(myFile);
    For I:=1 to N do Writeln(myFile,I);
    CloseFile(myFile);
    Reset(myFile);
    For I:=1 to N do
        Begin
            ReadLn(myFile,J);Write(J:5)
        End ;
    CloseFile(myFile);
    Append(myFile);
    For I:=N to N*2 do WriteLn(myFile,I);
    CloseFile(myFile);
    Reset(myFile);
    Repeat ReadLn(myFile,I);WriteLn(I:5);
    Until Eof(myFile);
    CloseFile(myFile);
End.
```



**Пример (Delphi,Lazarus)**

Программа позволяет ввести два числа, записать эти числа и сумму их в файл, потом считать записанные числа из файла и вывести их на экран.

```
Var myFile:TextFile;
      A,B,A1,B1,C1:Integer;

Begin
  AssignFile(myFile,'Ex1.dat');
  Rewrite(myFile);
  ReadLn(A, B);
  Write(myFile,A:4);
  Write(myFile,B:4);
  Write(myFile,A+B:4);
  Writeln(myFile);
  CloseFile(myFile);
  Reset(myFile);
  ReadLn(myFile,A1,B1,C1);
  WriteLn(A1:5,B1:5,C1:5);
  CloseFile(myFile);

End.
```

**Пример (Delphi,Lazarus)**

Программа вводит имя текстового файла, создает на диске текстовый файл и записывает в него вводимый текст. Признаком окончания ввода – пустая строка (повторное нажатие Enter).

```
Type fil = TextFile;
var
  Fl : fil;
  Name : string[12];
  Txt : string;
begin
  Write('введите имя файла для записи текста >');
  Readln(Name);
  AssignFile(Fl,Name);
  Rewrite(Fl);
  Writeln('Вводите текст (в конце Enter) :');
  Repeat
    Write(':>');
    Readln(Txt);
    Writeln(Fl,Txt);
  Until Txt='';
  CloseFile(Fl);
end.
```