

Глава 9. ВВОД-ВЫВОД ДАННЫХ И ФАЙЛОВАЯ СТРУКТУРА

- Логический и физический файлы
- Типы файловой переменной
- Общие процедуры работы с файлами
- Текстовые файлы
- Обмен данными с консолью
- Типизированные файлы
- Нетипизированные файлы

ЛОГИЧЕСКИЙ И ФИЗИЧЕСКИЙ ФАЙЛ

Файловая система, реализуемая в Delphi Pascal, состоит из двух уровней: логических файлов и физических файлов.

Логический файл – это переменная одного из файловых типов, определенных в Delphi Pascal. Введенная файловая переменная может быть использована как средство общения с любым физическим файлом. Специальной процедурой AssignFile устанавливается, что объявленный логический файл будет служить средством доступа к конкретному физическому файлу.

Var

```
F : TextFile; {файловая переменная – лог. файл}
```

Begin

```
AssignFile(F, 'A:\text.txt');
```

```
... {связывается физический файл text.txt  
на диске A: с логическим файлом F}
```

End.

Физический файл – это:

- именованная область на устройствах внешней памяти (адресная часть в виде строковой константы оформляется в соответствии с правилами Windows – 'C:\PASCAL\program.pas');
- логическое устройство.

ЛОГИЧЕСКИЕ УСТРОЙСТВА, БУФЕРИЗАЦИЯ

Логические устройства используются для обмена информацией с основными устройствами ввода-вывода, такими как дисплей, клавиатура, и т.п. Они имеют стандартные имена (также записываются в виде строковой константы):

'CON' – консоль (клавиатура и экран);

'LPT1', 'LPT2'... – параллельные порты (типа Centronix);

'PRN' – принтер (синоним имени LPT1);

'COM1', 'COM2'... – последовательные порты;

'NUL' – фиктивное устройство (пустой файл), используется для отладки.

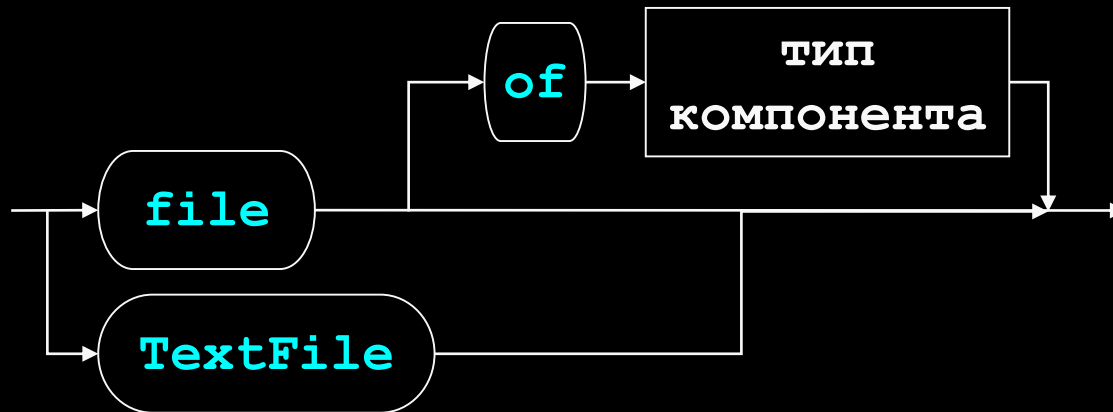
Физически операции ввода-вывода с файлами выполняются с использованием специального буфера. Стандартный объем буфера 128 байт. Физическая запись на устройство происходит только тогда, когда информацией будет занят последний байт буфера. Этим достигается компромисс между количеством и длительностью обращений к диску. **Буферизация** – накопление данных с целью обмена ими между программой и внешними устройствами.



ФАЙЛОВЫЕ ТИПЫ

Turbo Pascal поддерживает три файловых типа:

- текстовые файлы – **Var F1 : TextFile;**
- типизированные файлы – **Var F2 : file of Real;**
- нетипизированные файлы – **Var F3 : file;**

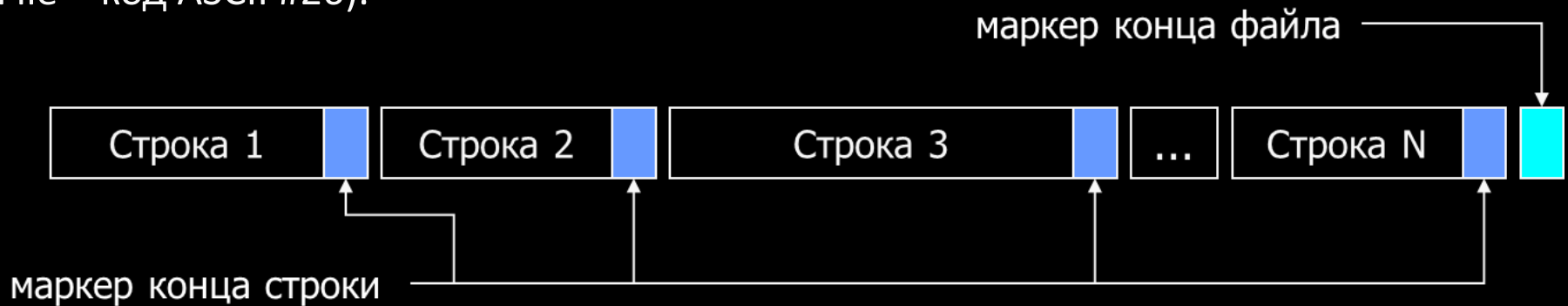


ОБЩИЕ ПРОЦЕДУРЫ РАБОТЫ С ФАЙЛОВЫМИ ПЕРЕМЕННЫМИ

- **AssignFile (Var F : <файловый тип>, St : String).** Инициализирует файловую переменную F, связывая ее с физическим файлом, определенным строкой St.
- **Reset (Var F : <файловый тип>).** Открывает файл, определенный файловой переменной F, для чтения. Для корректного выполнения процедуры необходимо иметь файл с соответствующим именем на диске.
- **Rewrite (Var F : <файловый тип>).** Открывает файл, определенный файловой переменной F, для записи. При открытии для записи существующего файла старый файл уничтожается.
- **AppEnd (Var F : TextFile).** Открывает текстовый файл, определенный файловой переменной F, для добавления строк.
- **Read (F,...), Readln(F,...), Write(F,...), Writeln(F,...).** Выполняют операции чтения/записи (кроме нетипизированных файлов).
- **CloseFile (Var F : <файловый тип>).** Закрывает файл, открытый для записи или чтения. Связь файла с файловой переменной сохраняется.

ТЕКСТОВЫЕ ФАЙЛЫ

Физический текстовый файл трактуется в Delphi Pascal как совокупность строк переменной длины, состоящих из символов кодовой таблицы. В конце каждой строки ставится маркер конца строки EOLN (End Of Line – последовательность кодов ASCII #13 и #10), а в конце всего файла – маркер конца файла EOF (End Of File – код ASCII #26).



Физические текстовые файлы связываются с файловыми переменными, принадлежащими типу `TextFile`.

Для ввода и вывода информации в текстовые файлы используют процедуры `Read` (`ReadLn`) и `Write` (`WriteLn`), первым параметром которых является имя логического файла, после чего следует список переменных, относящихся к символьному (`Char`) или строковому (`String`) типу, а также к любому целому или вещественному.

```
WriteLn(Var F : TextFile; X1, X2, ..., Xn) ;
```

ЧТЕНИЕ ИНФОРМАЦИИ ИЗ ТЕКСТОВОГО ФАЙЛА (READ И READLN)

- Если переменные ввода относятся к символьному типу Char, то символы считываются поочередно в соответствующие переменные (без разделителей).
- Для строкового типа String[N] количество "заносимых" символов в строковую переменную зависит от ее объявленной длины. Но если попался символ конца строки #13, то чтение строки прекращается.
- При введении числовых значений два числа считаются разделенными, если между ними есть пробел, символ табуляции (#9) или символ конца строки (#13). Если нарушен формат, то фиксируется ошибка ввода-вывода.

Процедура ReadLn считывает значения в текущей строке, и затем переводит позицию в начало следующей строки, даже если в текущей строке остались непрочитанные данные.

```
Var
    X1, X2 : Real;
    F : TextFile;      {F → 12.3 13.4 14.4 15.6}
Begin
    ...
    ReadLn (F, X1, X2);  {X1 → 12.3; X2 → 13.4}
End.
```

ЗАПИСЬ ИНФОРМАЦИИ В ТЕКСТОВЫЙ ФАЙЛ (WRITE И WRITELN)

При записи информации в текстовый файл список вывода может включать выражения типа Char, String, Boolean, а также целого или вещественного типов. Процедура Write выводит данные в текущую строку и не закрывает ее, т.е. следующие данные запишутся в ту же строку. Процедура WriteLn выводит строку данных и закрывает ее, приписывая символы #13 и #10 в ее конец. Имеется возможность управлять форматом вывода чисел. Список вывода может содержать не только переменные, но и константы, выражения, вызовы функций.

```
Write(F, RealVar, 'номер' , RealVar + Cos(5*5) ) ;
```

Обмен данными с консолью

В модуле System определены две переменные Input и Output типа TextFile, которые заранее связаны с физическим файлом-устройством 'CON' – консолью, которое объединяет клавиатуру и дисплей.

Если в процедурах ввода Read (ReadLn) опущено имя файла, то считается, что ввод идет из системного файла Input, что физически означает набор на клавиатуре. Аналогично, если в процедурах вывода данных Write (WriteLn) опущено имя файла, то они выводятся через файловую переменную Output на экран.

```
Write(Output, 23) = Write(23)
```


ТИПИЗИРОВАННЫЕ ФАЙЛЫ

Типизированный файл – это файл, все компоненты которого одного типа, заданного при объявлении файловой переменной. Тип данных может быть любым, кроме файлов, объектов и структурированных компонентов (массивов, записей, и т.д.), содержащих файлы или объекты. Компоненты файла хранятся на диске во внутреннем (двоичном) формате.



Так как каждый компонент занимает в файле одинаковый объем (равный размеру его типа), имеется возможность организовать прямой доступ к каждому из них. Процедура позиционирования

Seek (Var F : <файловый тип>; <номер компонента>)

смещает указатель файла к требуемому компоненту.

Ввод-вывод данных осуществляется стандартными процедурами Read, Write, параметрами которых кроме файловой переменной являются переменные такого же типа, как и тип файла.

НЕТИПИЗИРОВАННЫЕ ФАЙЛЫ

Нетипизированные файлы также состоят из машинных представлений данных, которые объявлены без указания типа его компонентов. Нетипизированный файл – это высокоскоростной низкоуровневый канал ввода-вывода для доступа к любым файлам с любым типом. С его помощью можно записывать на диск произвольные участки рабочей памяти компьютера и считывать их в память.

Операции чтения и записи осуществляется блоками с помощью процедур BlockRead и BlockWrite.