Режим просмотра файла

Если файл уже создан, то его можно просмотреть начиная с 1-го элемента, выполнив следующие действия:

1) Открыть файл для чтения

reset(f) **где f – логическое имя файла*

С помощью этой процедуры файл переводится в режим чтения и окно устанавливается на 1-ую позицию файла.

Эта процедура может быть применима к файлу любое число раз, при этом значение файла не изменяется.

2) Чтение записи (элемента) из файла

Осуществляется с помощью процедуры

read(f, a) **где f – логическое имя файла*,

a – переменная, в которую будет записан элемент.

Таким образом, в переменную А будет записан тот элемент файла, на котором сейчас установлено окно. После записи окно будет перемещено на следующий элемент.

Режим просмотра файла

Если сделана попытка открыть несуществующий файл, то возникает ошибка, приводящая к прекращению выполнения программы.

Этого можно избежать предварительно проверив существование файла с помощью стандартной функции IORESULT, которая возвращает 0, если файл существует или же число отличное от 0, если файл не существует, при этом сам контроль ошибок отключается функцией {$I-}, а обратно включается функцией {$I+}.

Пример:

```
{$I-} Reset (f); {$I+}
If IORESULT <> 0 then <файл не существует> else <файл существует>
```

Режим просмотра файла

При чтении элемента из файла или при открытии для чтения пустого файла может быть достигнут конец файла, т.е. окно выходит за пределы последнего элемента файла.

Для определения этого состояния в Pascal существует стандартная логическая функция *EOF (end of file – конец файла).*

Обращение к которой осуществляется с помощью указателя функции

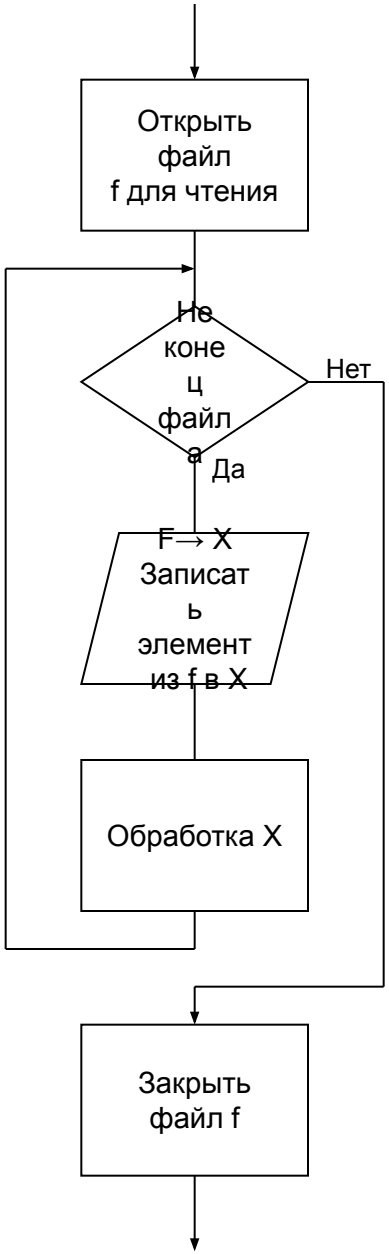
eof(f) **где f – логическое имя файла*

`Функция EOF(f)` возвращает *TRUE*, если конец файла **достигнут** или же *FALSE*, если конец файла еще **не достигнут.**

Режим просмотра файла заканчивается процедурой

close(f) **где f – логическое имя файла*

Алгоритм просмотра файла




```
Assign(f,'…');
Reset (f);
While not eof(f) do
Begin
    read(f,x);
    <обработка x>;
End;
Close(f);
……
```

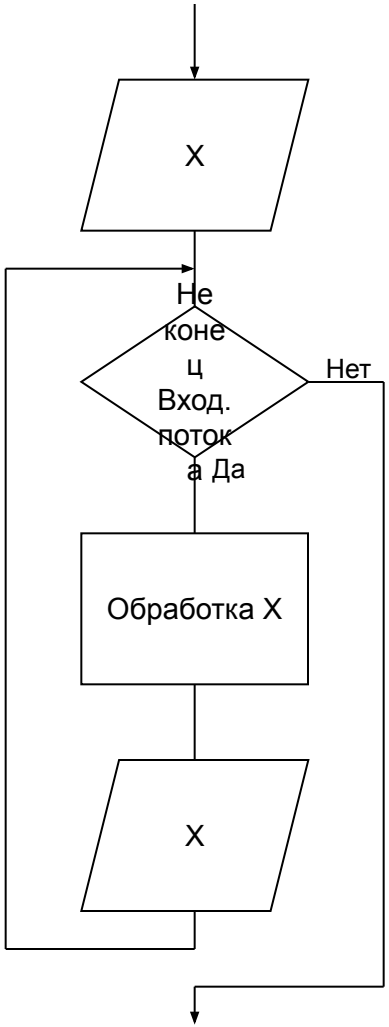
Ввод данных с клавиатуры

Рассматривается как входной файл со стандартным именем INPUT (входной поток)

Если неизвестно число вводимых данных, то признаком конца ввода может быть конец файла. Во входном потоке конец файла имитируется после нажатия клавиши Ctrl + Z или F6. При этом в начале программы необходимо присвоить переменной checkeof значение TRUE

```
Checkeof:=true;
```

Ввод данных с клавиатуры




```
Checkeof:=true;

……

Read(x);

While not eof do

Begin

    read(x);

End;

……
```

Если создается файл из входного потока, то для очистки буфера ввода необходимо выполнить эти 2 процедуры:

`close(input);` (очистить буфер ввода)

`reset(input);` (открыть входной поток для ввода данных)

Добавление элементов в файл

В Turbo Pascal разрешается открыть файл для чтения и записи. Это удобно использовать для добавления новых записей в конец файла.

```
Reset(f);
While not eof(f) do begin
    read(f,x);
    write(x);
End;
While not eof do begin
    read(x);
    write(f,x);
end;
Close(f);
```

Изменение элементов файла

Для изменения или удаления существующих элементов файла нужно создать дополнительный файл для записи в него элементов текущего файла.

Вот пример простейшей реализации редактирования:

1. Копирование элементов из текущего файла в дополнительный, а затем отчистка исходного файла.
2. По одному элементу у пользователя спрашивается о необходимости изменить (удалить) элемент файла.
3. В случае необходимости – элемент редактируется и записывается в изначальный файл, если же элемент не нужно редактировать, то он просто записывается в файл. И так будет происходить пока цикл не дойдет до конца дополнительного файла.
4. Оба файла закрываются.

Дополнительные стандартные функции и процедуры

1). Уничтожение файла

`erase(f);` **где f – логическое имя файла*

Для выполнения этой функции файл должен быть закрыт, а так-же необходима проверка на существование файла (которую мы рассматривали ранее)

2). Переименовывание файла

`erase(f,'новое имя');` **где f – логическое имя файла*

3). Смещение указателя (окно) в файле

`seek(f,n);` **где f – логическое имя файла*

n – выражение типа longint, которое указывает номер элемента в файле

4). Определение размера файла

`filesize(f);` **где f – логическое имя файла*

Должно быть присвоенно переменной типа longint

Дополнительные стандартные функции и процедуры

5). Определение номера элемента, на котором установлен указатель (окно) файла.

`filepos(f);` **где f – логическое имя файла*

Должно быть присвоено переменной типа longint