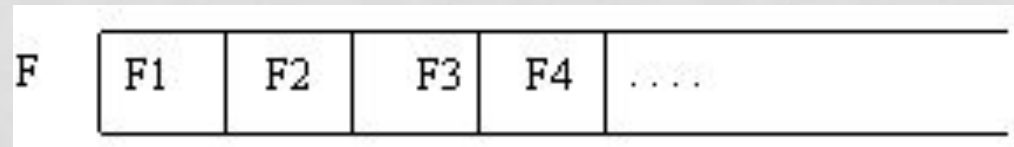


# РАБОТА С ФАЙЛАМИ



# ФАЙЛ

- Файл в Паскале можно изобразить как некоторую ленту, у которой есть начало, а конец не фиксируется. Элементы файла записываются на эту ленту последовательно друг за другом:



- где F – имя файла, а F1, F2, F3, F4 – его элементы.
- Файл во многом напоминает магнитную ленту, начало которой заполнено записями, а конец пока свободен.

# ФАЙЛ

- Важной особенностью файлов является то, что данные, содержащиеся в файле, переносятся на внешние носители.
- Файловый тип Паскаля – это единственный тип значений, посредством которого данные, обрабатываемые программой, могут быть получены извне, а результаты могут быть переданы во внешний мир. Это единственный тип значений, который связывает программу с внешними устройствами ЭВМ.

# ФАЙЛ ПОСЛЕДОВАТЕЛЬНОГО ДОСТУПА.

- У последовательного файла доступен всегда лишь очередной элемент.
- Если в процессе решения задачи необходим какой-либо из предыдущих элементов, то необходимо вернуться в начало файла и последовательно пройти все его элементы до нужного

# ФАЙЛЫ ПРОИЗВОЛЬНОГО ДОСТУПА

- ПОЗВОЛЯЮТ ВЫЗЫВАТЬ КОМПОНЕНТЫ В ЛЮБОМ ПОРЯДКЕ ПО ИХ НОМЕРУ.

# РАБОТА С ФАЙЛАМИ В ПАСКАЛЕ

- 3 характерные особенности любого файла.
  1. У него есть имя
  2. Он содержит компоненты одного типа
  3. Длина вновь создаваемого файла никак не оговаривается при его объявлении и ограничивается только емкостью устройств внешней памяти.

# 3 СПОСОБА СОЗДАНИЯ ПЕРЕМЕННОЙ ФАЙЛОВОГО ТИПА

1. Type <имя\_ф\_типа>=file of<тип\_элементов>;
2. <имя\_ф\_типа>=text;
3. <имя\_ф\_типа>=file;

- Здесь <имя\_ф\_типа> – имя файлового типа (правильный идентификатор); File, of – зарезервированные слова (файл, из); <тип\_элементов> – любой тип Паскаля, кроме файлов.

- В зависимости от способа объявления можно выделить три вида файлов Паскаля:
- • типизированные файлы Паскаля (задаются предложением `file of..`);
- • текстовые файлы Паскаля (определяются типом `text`);
- • нетипизированные файлы Паскаля (определяются типом `file`).



- Переменные файлового типа используются в программе только в качестве параметров собственных и стандартных процедур и функций.

# ОСНОВНЫЕ ПРОЦЕДУРЫ И ФУНКЦИИ ДЛЯ РАБОТЫ С ФАЙЛАМИ

- 1. До начала работы с файлами в Паскале необходимо установить связь между файловой переменной и именем физического дискового файла:
- `Assign(<файловая_переменная>, <имя_дискового_файла>)`

# ОСНОВНЫЕ ПРОЦЕДУРЫ И ФУНКЦИИ ДЛЯ РАБОТЫ С ФАЙЛАМИ

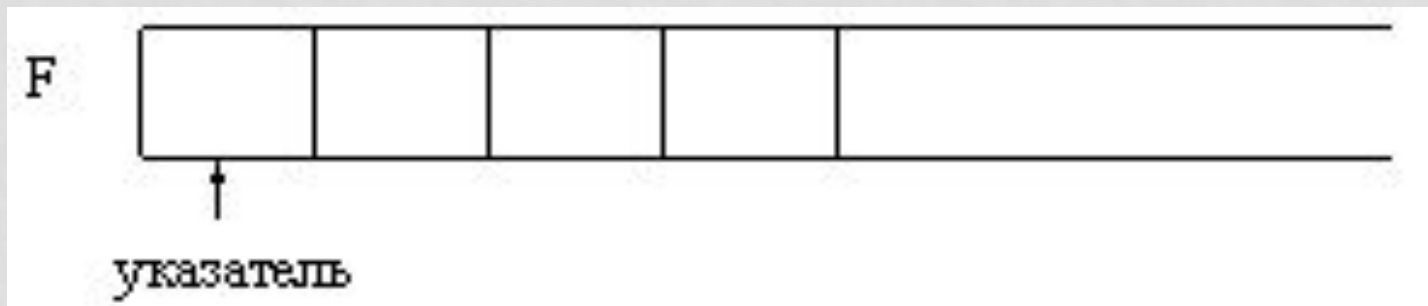
- Следует помнить, что имя дискового файла при необходимости должно содержать путь доступа к этому файлу, включая имя дисководов. При этом имя дискового файла – строковая величина, т.е. должна быть заключена в апострофы. Например:
- `Assign (chf, 'G:\Home\ Student\ Lang\ Pascal\ primer.dat');`

# ОСНОВНЫЕ ПРОЦЕДУРЫ И ФУНКЦИИ ДЛЯ РАБОТЫ С ФАЙЛАМИ

- 2. После окончания работы с файлами на Паскале, они должны быть закрыты.
- Close(<список файловых переменных>);

# ОСНОВНЫЕ ПРОЦЕДУРЫ И ФУНКЦИИ ДЛЯ РАБОТЫ С ФАЙЛАМИ

- 3. Подготовка к записи в файл Паскаля
- `Rewrite(<имя_ф_переменной>);`
- 
- Результат выполнения процедуры `rewrite(f);` выглядит следующим образом:



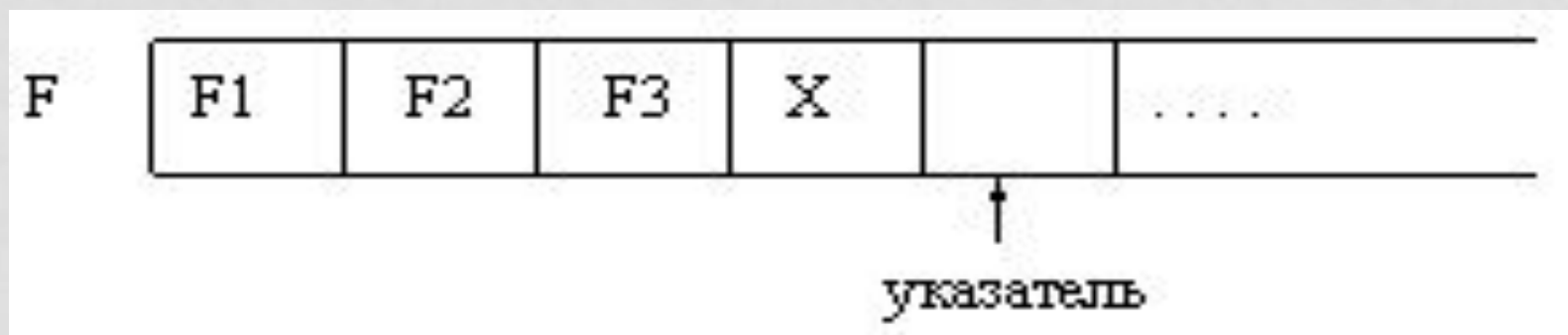
# ОСНОВНЫЕ ПРОЦЕДУРЫ И ФУНКЦИИ ДЛЯ РАБОТЫ С ФАЙЛАМИ

- **4.Запись в файл Паскаля**
- `Write(<имя_ф_переменной>, <список записи>);`
- Результат действия процедуры `write(f, x)` можно изобразить так:
- Состояние файла `f` до выполнения процедуры



# ОСНОВНЫЕ ПРОЦЕДУРЫ И ФУНКЦИИ ДЛЯ РАБОТЫ С ФАЙЛАМИ

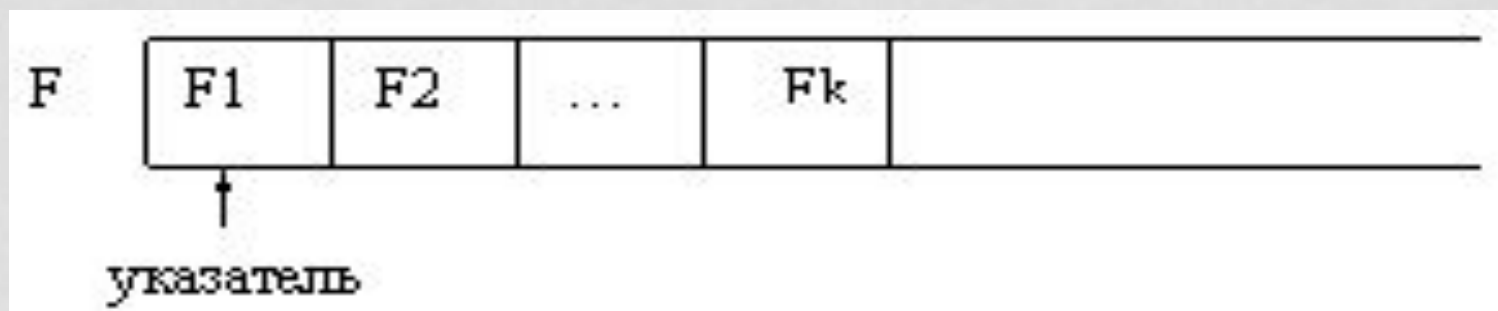
- Состояние файла  $f$  после выполнения процедуры



- Для типизированных файлов выполняется следующее утверждение: если в списке записи перечислено несколько выражений, то они записываются в файл, начиная с первой доступной позиции, а указатель смещается на число позиций, равное числу записываемых выражений.

# ОСНОВНЫЕ ПРОЦЕДУРЫ И ФУНКЦИИ ДЛЯ РАБОТЫ С ФАЙЛАМИ

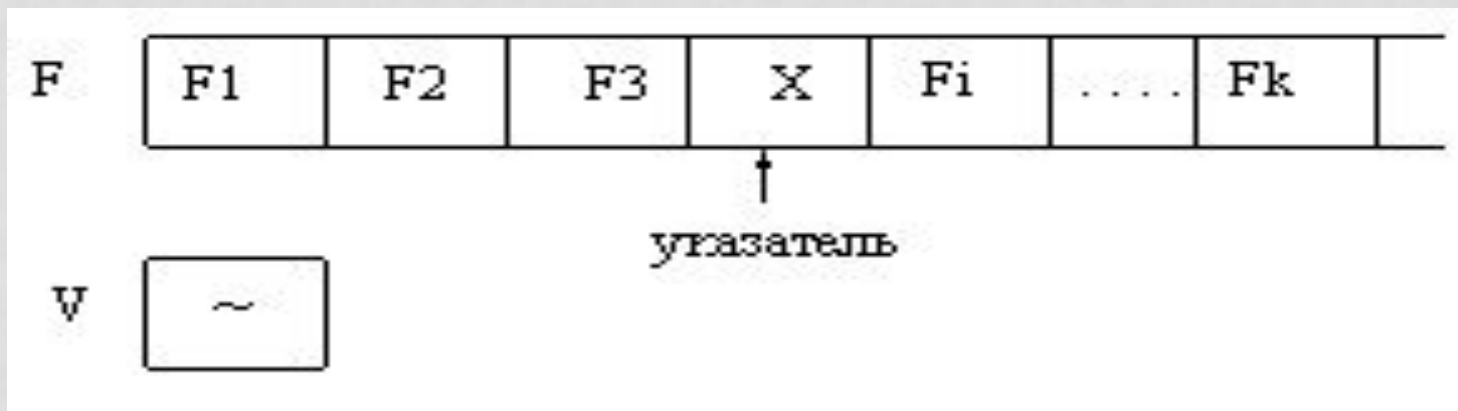
- 5. Подготовка файла к чтению Паскаля
- `Reset(<имя_ф_переменной>);`
- Результат выполнения этой процедуры можно изобразить следующим образом:





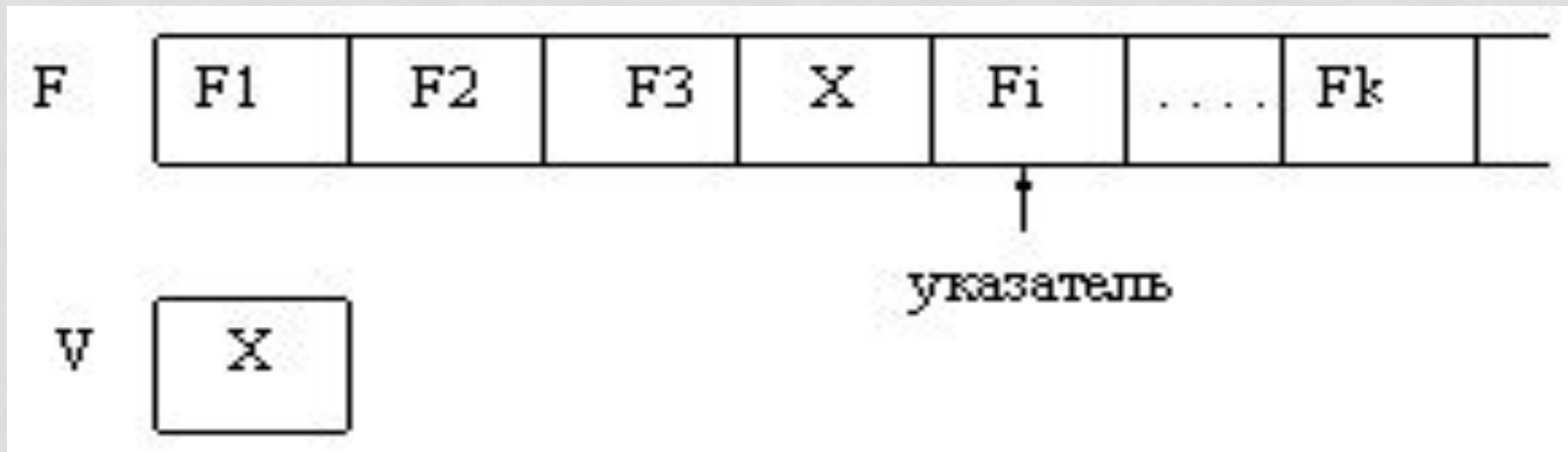
# ОСНОВНЫЕ ПРОЦЕДУРЫ И ФУНКЦИИ ДЛЯ РАБОТЫ С ФАЙЛАМИ

- 6.Чтение из файла в Паскале
- `Read(<имя_ф_переменной>,<список переменных>);`
- Рассмотрим результат действия процедуры `read(f, v)`:
- Состояние файла `f` и переменной `v` до выполнения процедуры:



# ОСНОВНЫЕ ПРОЦЕДУРЫ И ФУНКЦИИ ДЛЯ РАБОТЫ С ФАЙЛАМИ

- Состояние файла  $f$  и переменной  $v$  после выполнения процедуры:



# ОСНОВНЫЕ ПРОЦЕДУРЫ И ФУНКЦИИ ДЛЯ РАБОТЫ С ФАЙЛАМИ

- 7. Функция определения достижения конца файла в Паскале
- Eof(<имя\_ф\_переменной>);

# ОСНОВНЫЕ ПРОЦЕДУРЫ И ФУНКЦИИ ДЛЯ РАБОТЫ С ФАЙЛАМИ

- 8.Изменение имени файла в Паскале
- `Rename(<имя_ф_переменной>, <новое_имя_файла>);`

# ОСНОВНЫЕ ПРОЦЕДУРЫ И ФУНКЦИИ ДЛЯ РАБОТЫ С ФАЙЛАМИ

- **9.Уничтожение файла в Паскале**
- Erase(<имя\_ф\_переменной>);

# ОСНОВНЫЕ ПРОЦЕДУРЫ И ФУНКЦИИ ДЛЯ РАБОТЫ С ФАЙЛАМИ

- **10. Уничтожение части файла от текущей позиции указателя до конца в Паскале**
- `Truncate(<имя_ф_переменной>);`

# ОСНОВНЫЕ ПРОЦЕДУРЫ И ФУНКЦИИ ДЛЯ РАБОТЫ С ФАЙЛАМИ

- **11.Файл Паскаля может быть открыт для добавления записей в конец файла**
- `Append(<имя_ф_переменной>);`

- Типизированные файлы Паскаля. Длина любого компонента типизированного файла строго постоянна, т.к. тип компонент определяется при описании, а, следовательно, определяется объем памяти, отводимый под каждую компоненту.
- Это дает возможность организовать прямой доступ к каждой компоненте (т.е. доступ по порядковому номеру).



- Для перемещения указателя по файлу и доступа к компонентам типизированного файла существуют специальные процедуры и функции:
- **fileSize** (<имя\_ф\_переменной>) – функция Паскаля, определяющая число компонентов в файле;

- **filePos**(<имя\_ф\_переменной>) – функция Паскаля, значением которой является текущая позиция указателя;
- **seek**(<имя\_ф\_переменной>,n) – процедура Паскаля, смещающая указатель на компоненту файла с номером n. Так, процедура `seek(<имя_ф_переменной>,0)` установит указатель в начало файла, а процедура `seek(<имя_ф_переменной>, FileSize(<имя_ф_переменной>))` установит указатель на признак конца файла.