

# Глава 8 Работа с файлами

## 8.1 Файловая система

**Файл** – поименованная последовательность элементов данных (компонентов файла), расположенных, как правило, во внешней памяти.

Полное имя файла:

<Имя диска>:<Список имен каталогов>\<Имя файла>.<Расширение>

Имя файла в Windows составляют из строчных и прописных букв латинского и русского алфавитов, арабских цифр и некоторых специальных символов, например, символов подчеркивания «\_» или знаков доллара «\$»

Расширение определяет тип хранящихся данных, например:

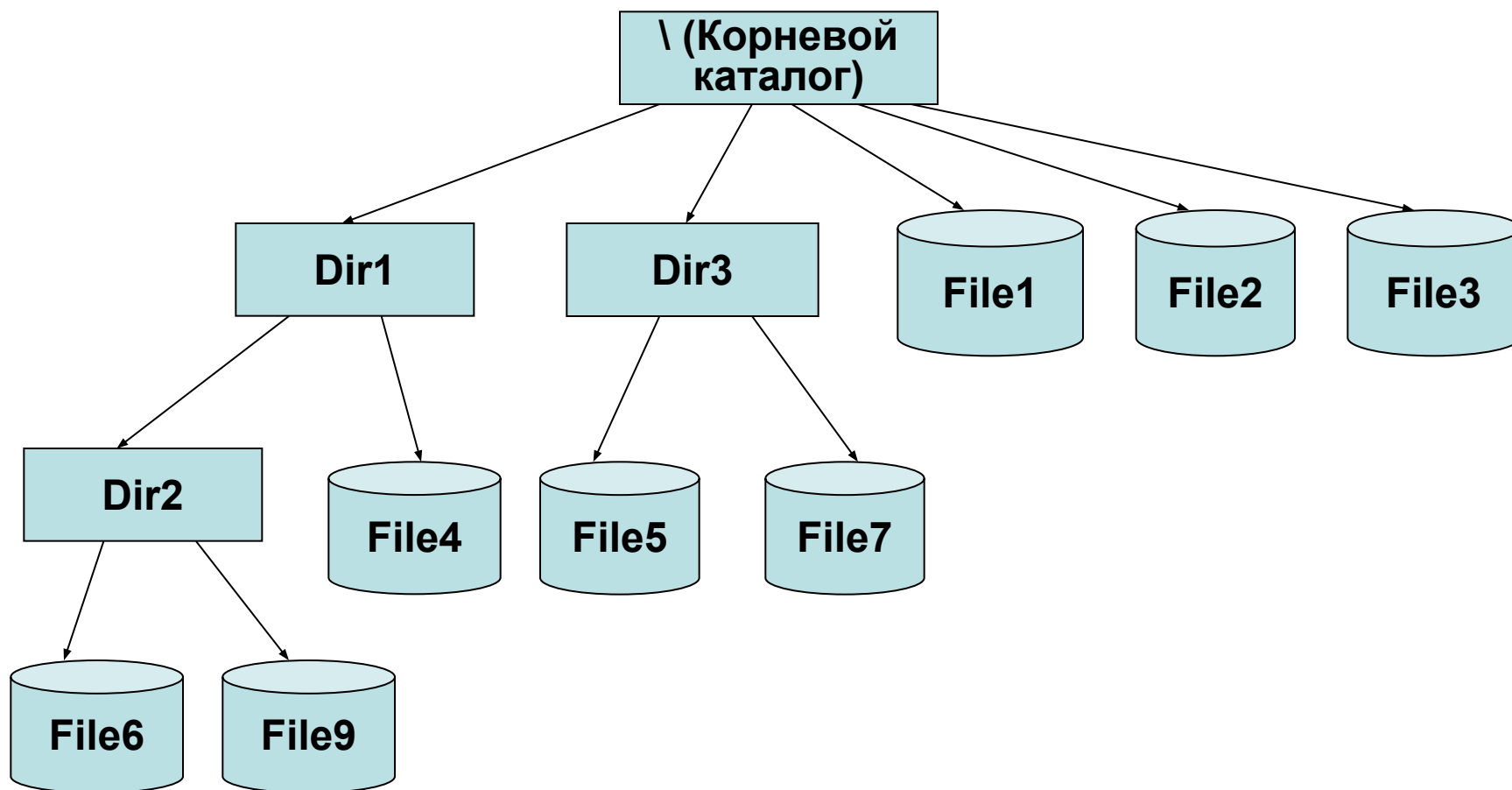
**COM, EXE** – исполняемые файлы (программы);

**PAS, BAS, CPP** – исходные тексты программ на алгоритмических языках ПАСКАЛЬ, БЭЙСИК и С++;

**BMP, JPG, PIC** – графические файлы (рисунки, фотографии);

**WAV, MP3, WMA** – музыкальные файлы.

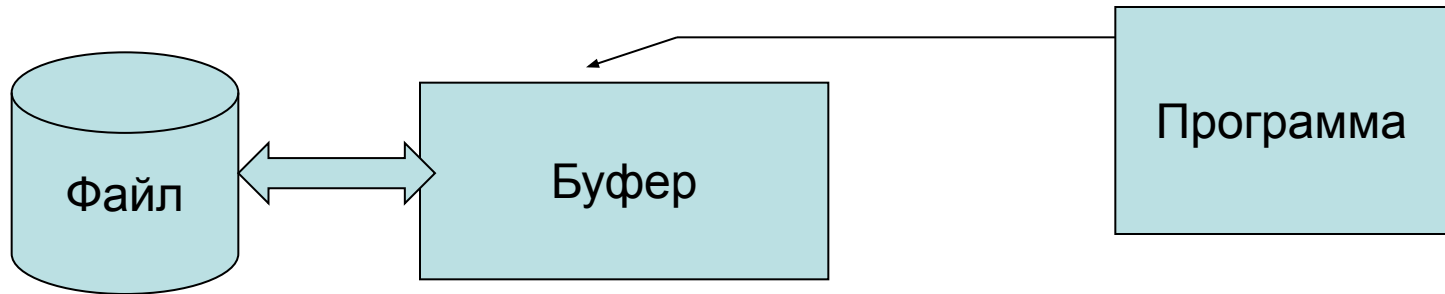
# Организация файлов на внешнем носителе



Пример:

D:\Dir1\Dir2\File9.pas

## 8.2 Механизм выполнения ввода-вывода



**typedef struct {**

**short level;** // количество непрочитанных байт в буфере

**unsigned flags;** // режимы: чтение/ запись/ввод/вывод и т.д.

**char fd;** // дескриптор файла

**unsigned hold;**

**short bsize;** // размер буфера

**unsigned char \*buffer;** // адрес буфера

**unsigned char \*curp;** // текущий указатель (в буфере)

**unsigned istemp;**

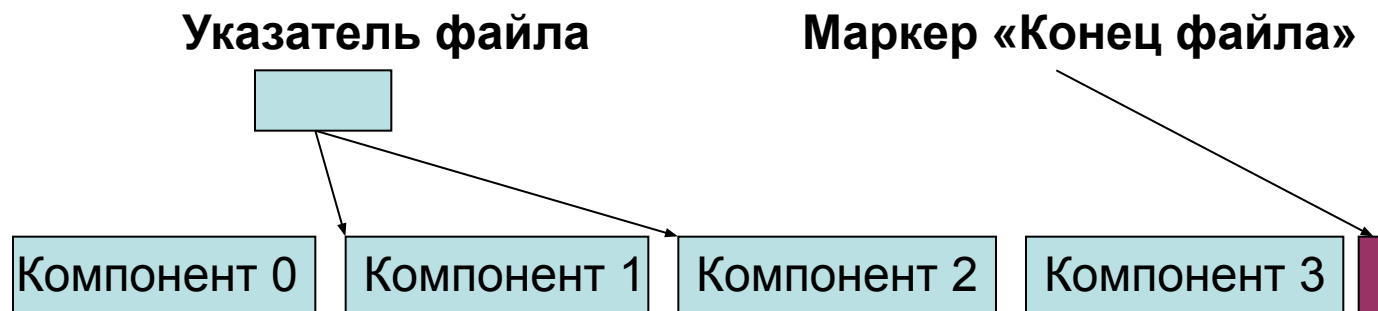
**short token;** } **FILE;**

Файловый указатель



# Указатель файла

Доступ к компонентам файла осуществляется через **указатель файла**. При выполнении операции чтения или записи указатель автоматически перемещается на следующий компонент.



После вывода последнего компонента файла система пишет специальную запись – **маркер «Конец файла»**.

При обнаружении во время операции чтения маркера конца файла – операция завершается. Попытка читать маркер вызывает прерывание по ошибке чтения.

## 8.3 Объявление, открытие и закрытие файлов

Объявление

**FILE \*<Файловая переменная (Ф.п.)>;**

файловая переменная является указателем на структуру, краткое название которой **FILE**

Работа с файлами включает:

- ***инициализацию файловой переменной*** – установление связи файловой переменной с файлом;
- ***открытие файла*** – подготовку файла для выполнения операций ввода-вывода;
- ***обработку компонентов файла*** – выполнение операций ввода-вывода;
- ***закрытие файла.***

# Объявление, открытие и закрытие файлов(2)

## Открытие

При открытии устанавливается связь файловой переменной с реальным файлом на диске (инициализация файловой переменной), а также указывается направление передачи данных: запись или чтение.

**<Ф. п.>=fopen(<Имя файла>, <Операция [+][Тип]>);**

**<Имя файла>** - символьное имя файла в соответствии с системными соглашениями. Если файл находится в текущем каталоге, то достаточно указать имя файла и его расширение. В противном случае необходимо указать полное имя файла, включающее имя диска и каталоги.

Для определения направления передачи данных и специфики обработки используется опция операция.

# Объявление, открытие и закрытие файлов(3)

**<Операция[+]>:**

**r** - ввод из существующего файла;

**w** - вывод с очисткой файла или создание нового файла для вывода;

**a** - добавление к существующему или создание файла для вывода;

**r+** - ввод/вывод в существующий файл;

**w+** - ввод/вывод в существующий или создание нового файла;

**a+** - ввод/добавление к существующему или создание файла для ввода/вывода.

**<Тип>** - **t** - текстовый файл (принимается по умолчанию);

**b** - двоичный файл.

Закрытие

**fclose(<Ф. п.>);**

# Обработка компонентов файла

Основные операции над компонентами – операции записи и чтения. На базе этих операций выполняют более сложные операции:

- *создание файла* – занесение в файл требуемых записей;
- *модификация файла* – изменение всех или нескольких записей, добавление и удаление записей;
- *поиск* нужной информации в файле.

Операции записи и чтения для каждого типа файла осуществляется по-своему.



# Примеры открытия/закрытия файлов

а) объявление и открытие существующего или нового двоичного файла для ввода/вывода

```
FILE *f;  
f=fopen("abc.txt", "w+b");  
...  
fclose(f);
```

б) объявление и открытие существующего файла с проверкой существования

```
FILE *f;  
if ((f=fopen("f:\\iva\\text.txt", "r")) != NULL) ...  
...  
fclose(f);
```

## 8.4 Функции управления файловым указателем

а) определение положения файлового указателя:

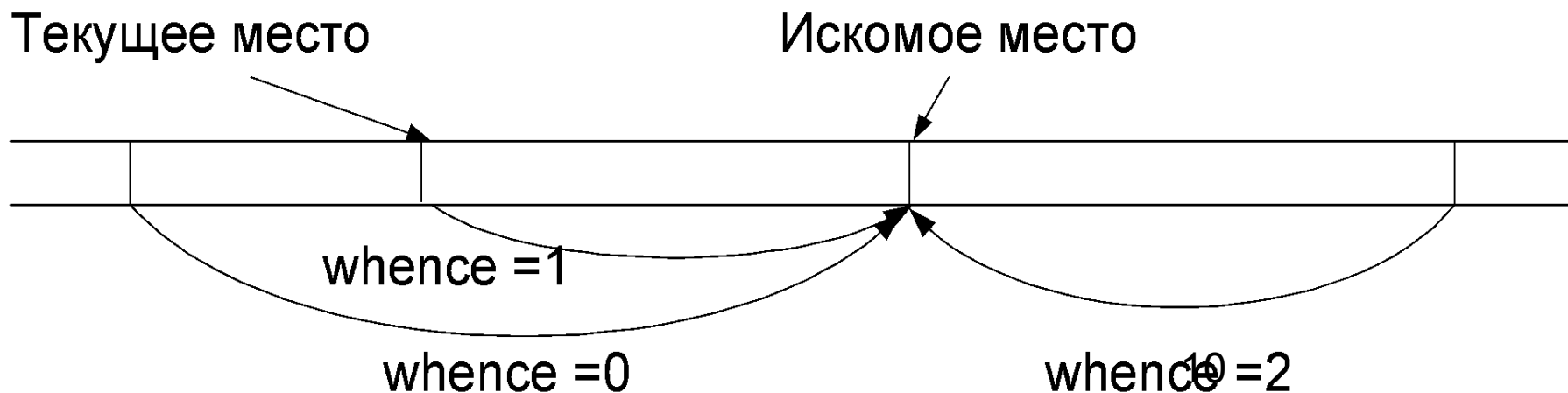
```
long ftell(FILE *stream);
```

б) установка файлового указателя на начало файла:

```
int rewind(FILE *stream);
```

в) установка файлового указателя в произвольное место:

```
int fseek(FILE *stream, long offset, int whence);
```



## 8.5 Текстовые файлы

1-2. Ввод/вывод символов

```
int getc(FILE *stream); // возвращает символ или EOF
int putc(int c,FILE *stream);
```

**Пример.** Вывод на экран содержимого файла (**Ex4\_01**).

```
#include <stdafx.h>
#include <stdio.h>
void main(int argc, char *argv[ ])
{ FILE *in;    int ch;
  if (argc<2) puts("Введите имя файла.");
  else
    if ((in=fopen(argv[1], "r")) !=NULL)
      { while ((ch=getc(in)) !=EOF) putchar(ch);
        fclose(in);
      }
    else puts("Нельзя открыть файл.");
}
```

# Текстовые файлы (2)

Стандартные текстовые файлы:

`stdin, stdout, stderr`

`getchar( ) == getc(stdin)`

`putchar(ch) == putc(ch, stdout)`

Пример. Чтение с начала и с конца (**Ex4\_02**).

```
#include "stdafx.h"
```

```
#include <stdio.h>
```

```
void main()
```

```
{ FILE *f; long offset=0L; int ch;
```

```
f=fopen("test.dat", "r");
```

```
while ((!fseek(f, offset++, 0)) && ((ch=getc(f)) != EOF))
```

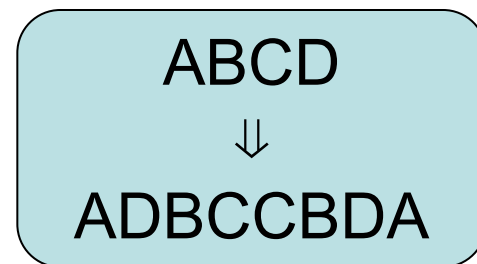
```
{ putc(ch, stdout);
```

```
if (!fseek(f, -(offset+2), 2)) putc(getc(f), stdout);
```

```
}
```

```
fclose(f);
```

```
}
```



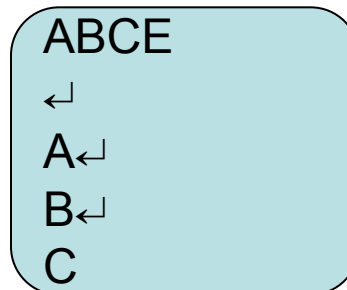
# Буферированные и «прямые» операции

**stdio.h:** `getchar()`, `putchar()` – буферированные;

**conio.h:** `getch()`, `getche()`, `putch()` – «прямые»

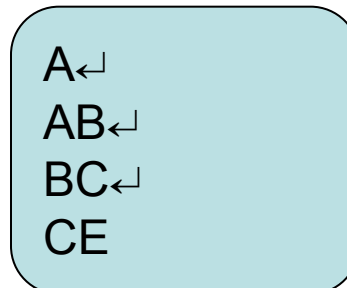
Примеры (Ex4\_03):

а) `while((n=getchar())!='E') {putchar('\n'); putchar(n);}`



```
ABCE
↵
A↵
B↵
C
E
```

б) `while ((n=getche())!='E') {putch('\n'); putch(n); }`



```
A↵
AB↵
BC↵
CE
```

# Текстовые файлы (3)

3. Вывод строк

```
int fputs(const char *s, FILE *stream);
```

Пример. Создание файла из 6 строк ([Ex4\\_04](#)).

```
#include "stdafx.h"
#include <stdio.h>
int main(int argc, char* argv[])
{ FILE *f;      int n;      char *s="ABCD";
  f=fopen("test.dat", "w");
  for (n=0;n<6;n++)
    { fputs(s, f);      fputs("\n", f); }
  fclose(f);
  return 0;
}
```

ABCD↵ABCD↵ABCD↵ABCD↵ABCD↵ABCD↵

## Текстовые файлы (4)

### 4. Ввод строк

`char *fgets(char *s, int n, FILE *stream);` // возвращает строку или NULL

Пример. Чтение файла по строкам ([Ex4\\_05](#)).

```
#include "stdafx.h"
#include <stdio.h>
void main()
{ FILE *f1;
  char string[80];
  f1=fopen("test.dat", "r");
  while (fgets(string, 80, f1) !=NULL)
    puts(string);
}
```

# Текстовые файлы(5)

5-6. Форматный ввод/вывод

```
int fscanf(FILE *stream, const char *format[, adress, ...]);  
int fprintf(FILE *stream, const char *format[, argument, .]);
```

Пример. Создание и распечатка файла чисел (**Ex4\_06**).

```
#include "stdafx.h"  
#include <stdio.h>  
#include <stdlib.h>  
#include <time.h>  
int main(int argc, char* argv[])  
{ int i,r; FILE *f;  
  srand((unsigned)time(NULL));  
  f=fopen("rand1.dat", "w+");  
  for (i=0;i<7;i++){r=rand(); fprintf(f,"%d ",r);}  
  rewind(f);  
  while (fscanf(f,"%d",&r)!=EOF)printf("%6d",r);  
  fclose(f); return 0;}
```

```
3684 20574 6789 23678 4578 9842  
31567
```



## 8.6 Двоичные файлы

Ввод/вывод

```
size_t fread(void *ptr, size_t size, size_t n,  
             FILE *stream) ;  
size_t fwrite(void *ptr, size_t size, size_t n,  
             FILE *stream) ;
```

Используются в двух вариантах:

а) при работе со структурами

```
fread (&myrec, sizeof(myrec), 1, f1) ;  
fwrite (&myrec, sizeof(myrec), 1, f1) ;
```

где `myrec` - переменная типа «структура»;

б) при работе с нетипизированной информацией

```
fread (&buffer, 1, sizeof(buffer), f2) ;  
fwrite (&buffer, 1, sizeof(buffer), f2) ;
```

где `buffer` – массив байт.

# Пример работы с двоичным файлом (1)

```
#include "stdafx.h" (Ex4_07)
#include <stdio.h>
#include <string.h>
struct toys{ char name[20];int cost;} toy;

int main(int argc, char* argv[])
{ FILE *f;
  f=fopen("test.dat","w+b");
  while (scanf ("\n%s",toy.name) ,
          strcmp (toy.name,"end") !=0)
  {   scanf ("%d",&toy.cost);
      fwrite (&toy,sizeof (toy) ,1 ,f) ;
  }
  fclose (f) ;
  return 0;
}
```

## Пример работы с двоичным файлом (2)

```
#include "stdafx.h" (Ex4_08)
#include <stdio.h>
struct toys{ char name[20]; int cost;} toy;

int main(int argc, char* argv[])
{ FILE *f;
  f=fopen("test.dat","r+b");
  while(fread(&toy,sizeof(toy),1,f)>0)
    printf("Toy name %s - cost - %d\n",
           toy.name, toy.cost);

  fclose(f);
  return 0;
}
```

# Чтение текстового файла как двоичного

```
#include "stdafx.h" (Ex4_09)
#include <stdio.h>

int main(int argc, char* argv[])
{ char c;
  FILE *f;
  f=fopen("ddd.dat", "w");
  fputs("ABCDEF", f);
  fclose(f);
  f=fopen("ddd.dat", "rb");
  while (fread(&c, 1, 1, f) != 0)
      printf("%c ", c);

  fclose(f);
  return 0;
}
```

ABCDEF

## 8.7 Переименование и удаление файлов

**Пример.** Вставка 10 чисел после первых 10 чисел файла.

```
#include "stdafx.h" (Ex4_10)
#include <stdio.h>
int main(int argc, char* argv[])
{ int n,m;
  FILE *f,*g;
  f=fopen("rand.dat","r");
  g=fopen("$$$$xxx.tmp","w");
  for (n=0;n<10;n++)
    { fscanf(f,"%d\n",&m); fprintf(g,"%d\n",m);}
  for (n=0;n<10;n++) fprintf(g,"%d\n",n);
  n=fgetc(f);
  while(n!=EOF) { fputc(n,g); n=fgetc(f);}
  fcloseall();
  unlink("rand.dat");
  rename("$$$$xxx.tmp","rand.dat");
  return 0;}
```