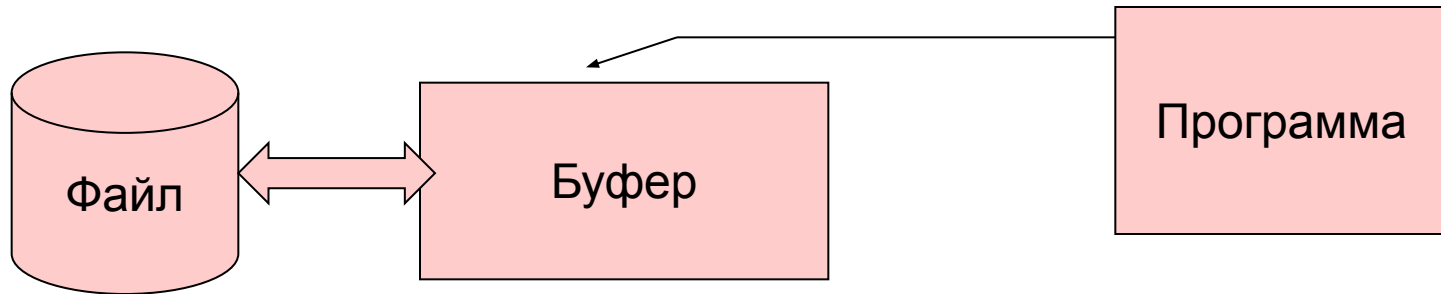


Глава 4. Работа с файловой системой

МГТУ им. Н.Э. Баумана
Факультет Информатика и системы управления
Кафедра Компьютерные системы и сети
Лектор: д.т.н., проф.
Иванова Галина Сергеевна

4.1 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;
```



4.2 Объявление, открытие и закрытие файлов

Объявление

FILE *<Файловая переменная (Ф.п.)>;

Открытие

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

<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.>);

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);
```

4.3 Файловый указатель.

Функции управления файловым указателем

а) определение положения файлового указателя:

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

б) установка файлового указателя на начало файла:

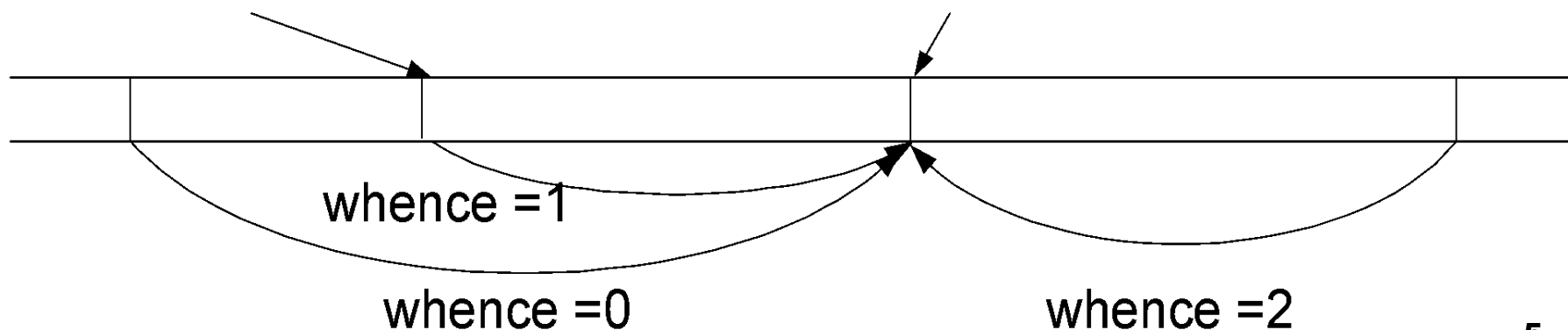
```
int rewind(FILE *stream);
```

в) установка файлового указателя в произвольное место:

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

Текущее место

Искомое место



4.4 Текстовые файлы

1-2. Ввод/вывод символов

```
int getc(FILE *stream); //возвращает символ или EOF(-1)
int putc(int c,FILE *stream);
```

Пример. Вывод на экран содержимого файла (**Ex4_01**), **полное** имя которого указывается в командной строке.

```
#include <stdio.h>

int main(int argc,char *argv[ ])
{ FILE *in;      int ch;
  if (argc<2)puts("Enter file name.");
  else
    if ((in=fopen(argv[1],"r"))!=nullptr)
      { while ((ch=getc(in))!=EOF) putchar(ch);
        fclose(in);
      }
    else puts("No open file.");
  return 0; }
```

Стандартные текстовые файлы

`stdin, stdout, stderr`

`getchar() == getc(stdin)`

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

Пример. Чтение с начала и с конца (**Ex4_02**) .

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

```
int main()
```

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

```
f=fopen("D:\\Ex2022\\Ex4_02\\test.dat", "r");
```

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

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

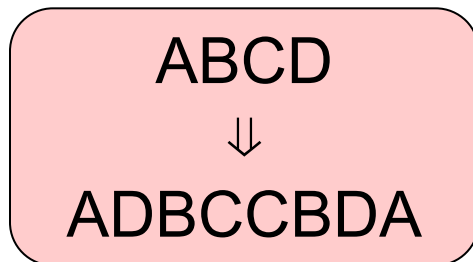
```
if (!fseek(f, -(offset+2), 2)) putc(getc(f), stdout);
```

```
}
```

```
fclose(f);
```

```
return 0;
```

```
}
```



Буферированные и «прямые» операции

`stdio.h`: `getchar()` , `putchar()` – буферированные операции;
`conio.h`: `getch()` – прямой ввод без копирования на экран,
`getche()` – тоже, но копирует на экран вводимый символ,
`putch()` – прямой вывод символа на экран.

Примеры (Ex4_03):

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

ABCE

↵

A↵

B↵

C

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

A↵

AB↵

BC↵

CE

3. Вывод строк

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

Пример. Создание файла из 6 строк (Ex4_04).

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

```
int main()
```

```
{    FILE *f;    int n;  
    const 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↵

4. Ввод строк

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

Пример. Чтение файла по строкам (**Ex4_05**).

```
#include <stdio.h>

int main()
{
    FILE *f1;
    char string[80];
    f1 = fopen("test.dat", "r");
    while (fgets(string, 80, f1) != nullptr)
        puts(string);
    return 0;
}
```

string

41	42	43	44	00	..	
A	B	C	D	\n	\0	

5-6. Форматный ввод/вывод

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

Пример. Создание и распечатка файла чисел (Ex4_06).

```
#include <stdio.h>  
#include <stdlib.h>  
#include <time.h>  
  
int main()  
{   int i,r; FILE *f;  
    srand(static_cast<unsigned int>(time(nullptr)));  
    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
```

Очистка буфера ввода

Пример (Ex4_11):

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

```
int main() {  
    int i;  
    char string[81];  
    printf("Enter with scanf: ");  
    for(i= 0;i<3;i++){  
        scanf("%s",string); // не вводит Enter!  
        printf("%s\n",string);  
    }  
    // Необходимо освободить буфер, содержащий код Enter !!!  
    while (getchar() != '\n');  
    printf("Enter gets: " );  
    gets(string);  
    printf("%s\n", string );  
}
```

```
Enter with scanf: rrr ttt yyy↵  
rrr  
ttt  
yyy  
Enter with gets: rrr ttt yyy↵  
rrr ttt yyy
```

4.5 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

Создание файла записей "Название игрушки, стоимость"
(Ex4_07)

```
#include <stdio.h>
#include <string.h>
struct toys{ char name[20];int cost;};
int main() {
    FILE *f;
    toys toy;
    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);
}
```

Пример работы с двоичным файлом (2)

Чтение и вывод на экран файла записей "Название игрушки, стоимость" (Ex4_08)

```
#include <stdio.h>

struct toys{ char name[20]; int cost;};

int main()
{
    FILE *f;
    toys toy;
    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

Создание текстового файла и чтение его как файла
символов (Ex4_09)

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

```
int main()
```

```
{ 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

4.6 Переименование и удаление файлов

Пример. Вставка 10 чисел после первых 10 чисел файла (Ex4_10)

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