

Работа с файлами

FILE

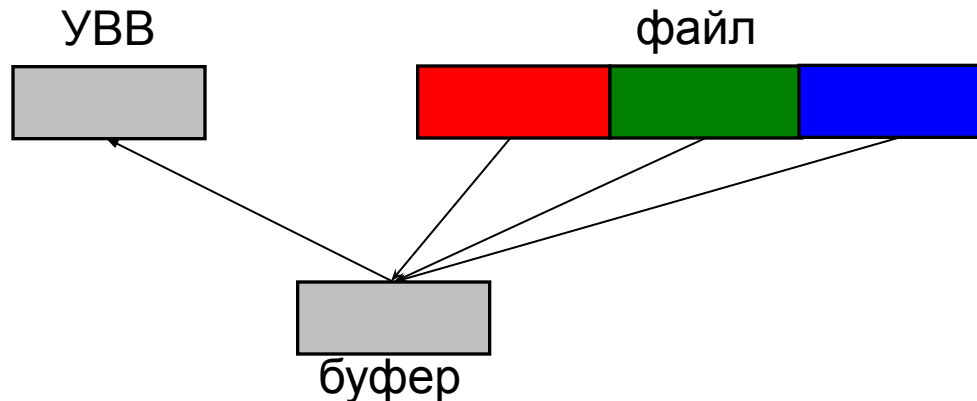
Ввод/вывод в C++

- функций библиотеки языка C **<stdio.h>**: `printf()`, `scanf()`, `gets()`, `puts()`, `getchar()`, `getche()`, `getc()`, `putchar()`;
- функций библиотеки языка C++ **<iostream.h>**: стандартный входной поток (объект `cin` типа `istream`), стандартный выходной поток (объект `cout` типа `ostream`) и операции для потока ввода `>>` и для потока вывода `<<`.

При этом данные:

- читаются из файлов и с устройств и пишутся в файлы и на устройства;
- передаются побайтно или группами байтов;
- рассматриваются как неструктурированный набор данных – поток.

Буфер ввода/вывода



```
void setbuf(FILE *f, char *p);
```

```
void setbuf(FILE *f, char *p, int mode, size_t size);
```

- буферизация ввода/вывода осуществляется автоматически и позволяет **ускорить выполнение программы** за счёт уменьшения количества обращений к медленно работающим внешним устройствам

Файловый ввод/вывод

- Включение файла заголовков `<stdio.h>` и объявления указателей файлов.
- Открытие файла вызовом функции `fopen`, присвоение возвращаемого ею значения указателю файла.
- Проверка правильности открытия файла (указатель файла после шага 2 не должен равняться `NULL`).
- Чтение и запись в файл с помощью функций `fprintf`, `fscanf`, `fputs`, `fgets`, `fgetc`, `fputc`, `getc`, `putc`, `fread`, `fwrite` и других, объявленных в библиотеке `<stdio.h>`.
- Закрытие файла вызовом функции `fclose`.

Функции для работы с файлами

Открытие потока

```
FILE *fopen (const char *name, const char *mode);
```

Заккрытие потока

```
int fclose (FILE *f);
```

Посимвольный ввод/вывод

```
int fgetc (FILE *f);
```

```
int fputc (int ch, FILE *f);
```

Ввод/вывод символьных строк

```
char* fgets (char *str, int n, FILE *f);
```

```
int fputs (char *str, FILE *f);
```

Форматированный ввод/вывод

```
int fprintf (FILE *f, const char *format, ...);
```

```
int fscanf (FILE *f, const char *format [, par1, par2, ...]);
```

Пример



```
#include <stdio.h>
#include <stdlib.h>
main()
{
    FILE *myfile;
    char file_name[40];
    gets(file_name);
    if ((myfile = fopen(file_name, "r")) == NULL)
    {
        printf("ошибка при открытии файла %s \n", file_name);
        exit(1);    //exit(0) -нормальное завершение процесса
    }            //exit(1) - сигнал об ошибке
    //работа с файлом с использованием myfile
    .....
    fclose(myfile);
}
```

Пример



```
#include <conio.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
void main()
{
    FILE *fi;
    char str[80];
    clrscr();
    if ((fi = fopen("file.dat", "w")) == NULL) exit(1);
    else while (strcmp(gets(str), " ") != 0) fputs(str, fi);
    fclose(fi);
    if ((fi = fopen("file.dat", "r")) != NULL)
        while (fgets(str, 80, fi) != NULL) puts(str);
    fclose(fi);
    getch();
}
```

Функции для работы с файлами

Ввод/вывод записей

```
size_t fread (void *buf, size_t size, size_t count, FILE *f);  
size_t fwrite(const void *buf, size_t size, size_t n, FILE *f);
```

Пример

```
struct book {  
    char title[81];  
    char author[21];  
    int page;  
    float price;  
};  
struct book library;  
FILE *fi = fopen("library.dat", "r");  
n = fread (&library, sizeof(library), 1, fi); //n=1, чтение 1 записи  
//в структуру library
```


Функции для работы с файлами

Очистка буфера

```
int fflush (FILE *f);
```

Работа с позицией в файле

```
int feof (FILE *f);
```

```
int fgetpos (FILE *f, fpos_t *pos);
```

```
int fseek (FILE *f, long k, int org);
```

```
int fsetpos (FILE *f, const fpos_t *pos);
```

```
long int ftell (FILE *f);
```

Работа с ошибками

```
int ferror (FILE *f);
```

```
int clearerr (FILE *f);
```

```
void rewind (FILE *f);
```

Переименование файла

```
int rename (const char *old_name, const char *new_name);
```

Пример

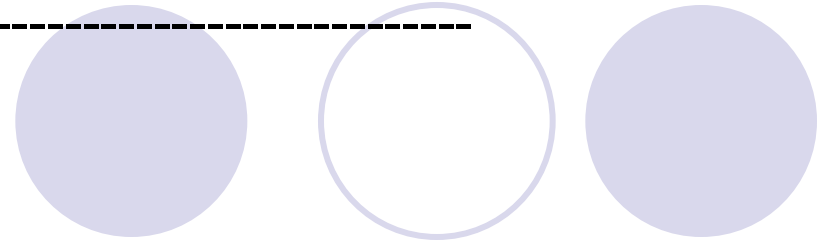


```
#include <conio.h>
#include <stdlib.h>
#include <ctype.h>
#include <stdio.h>
#define BYTES 7
FILE *fb;
int main()
{
    char file_name[30],buf[BYTES],str[40];
    int i,n,x;

    puts("Введите имя файла: ")
    while (fb=fopen(gets(file_name),"rb")==0)
    {
        printf("Ошибка открытия файла %s\n",file_name);
        putchar('\n');
    }
}
```

-----Вывод текстового файла-----

```
puts("ТЕКСТ: ");  
while (fgets(str,80,fb))  
    puts(str);  
puts("");
```

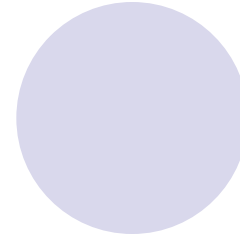
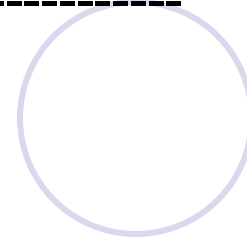
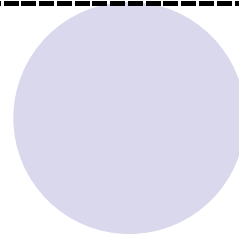


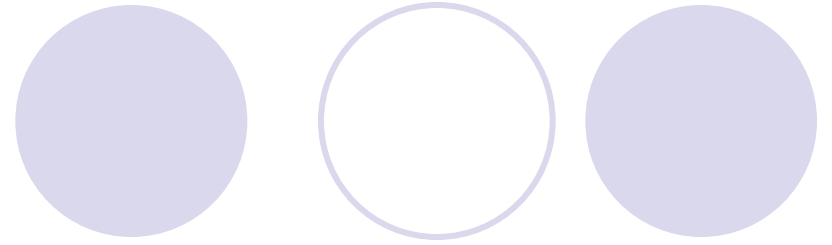
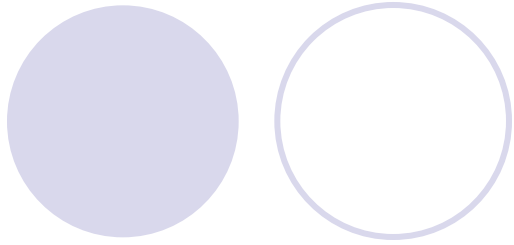
-----Вывод текста и чисел файла-----

```
puts("ТЕКСТ И ЧИСЛА: ");  
fseek(fb,0,0);  
fgets(str,80,fb);  
puts(str);  
for(i=0;i<5;i++)  
{  
    fscanf(fb,"%d",&x);  
    printf("%2d",x);  
}  
puts("");
```

-----Вывод бинарного файла-----

```
puts("БИНАРНЫЙ ФАЙЛ: ");
fseek(fb,0,0);
do
{
    n=fread(buf,1,BYTES,fb);
    puts("");
    for(i=0;i<n;i++) printf("%#5x",buf[i]);
    puts("");
    for(i=0;i<n;i++)
        if(iscntrl(buf[i])) printf("%5c",'.');
        else printf("%5c",buf[i]);
    puts("");
}
while(n==BYTES)
{
    getche();
    return 0;
}
getch();
}
```





Вопросы?