

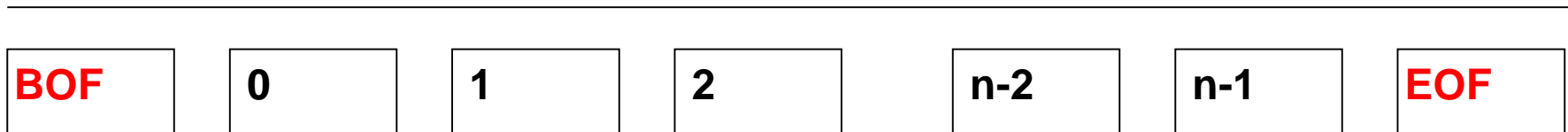
# ТЕМА 1.Файлы

---

## *1.1. Понятие файла*

**Текстовые файлы**

**Бинарные файлы**



BOF (Begin of File)

EOF (End of File)

## 1.2. Функции для работы с файлами

`stdio.lib` ( `#include <stdio.h>` )  
`io.lib` ( `#include <io.h>` ).

**FILE** *\*указатель на файл;*

**FILE** \*f11, \*f12;

**NULL**

**EOF**

**FOPEN\_MAX**

## *1.2.1. Функции для открытия-закрытия файла*

```
errno_t fopen_s( FILE **указатель_на_файл,  
                const char *имя_файла,  
                const char *режим_открытия);
```

**r**

**w**

**a**

**b**

**t**

**+**



```
FILE* fl;  
fopen_s(&fl, "lab1.dat", w");
```

```
FILE* fl;  
errno_t err = fopen_s(&fl, "lab1.dat", w");  
    if (err) {  
cout << "The file was not opened";  
    return 1;  
    }
```

```
FILE* fl;
    if (fopen_s(&fl, "f:\\lab1.dat", "r"))
{
    cout << "The file was not opened" << endl;
errno_t err = fopen_s(&fl, "f:\\lab1.dat", "w+");
    if (!err) cout << "The file was created" << endl;
    else cout << "The file was not created" << endl;
}
```

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

```
int _fcloseall(void);
```

```
if (fl)
{
err = fclose(fl);
if (!err) cout << "The file was closed" << endl;
else cout << "The file was not closed" << endl;
}
```

```
int n = _fcloseall();
cout << "Number of files closed "
      << n << endl;
```

## *1.2.1. Функции для модификации содержимого файла*

```
int putc(int СИМВОЛ,  
        FILE * указатель_на_файл);
```

```
int getc(FILE * указатель_на_файл);
```



# Пример: Запись/Чтение данных из файла.

```
char ch;  
do {  
    cin >> ch;  
    putc(ch, fl);  
} while (ch != '!');  
  
fclose(fl);  
fopen_s(&fl, "f:\\lab1.dat", "rt");  
while (ch = getc(fl) != EOF)  
    cout << (char) ch << " ";
```

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

**Пример:** Чтение данных из файла.

```
while( !feof(fl) ) cout << getc(fl) << " ";
```

```
int fputs(const char * строка,  
          FILE * указатель_на_файл);
```

```
char *fgets(char *строка, int длина,  
            FILE * указатель_на_файл);
```

```
int *fprintf(FILE * указатель_на_файл,  
             const char * управляющая_строка);
```

**%**[*флаг*][*ширина*][*.точность*][*размер*]*тип*

```
int *fscanf(FILE * указатель_на_файл,  
            const char * управляющая_строка);
```

## Пример.

Запись и чтение данных из файла.

```
double v=3.14; int n=298;
```

```
fprintf( fl, "%s %e %d %c", "string", v, n, 'x');
```

...

```
double v; char st[81]; int n; char ch;
```

```
fscanf(fl, "%s %e %d %c ", st, &v, &n, &ch);
```



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

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

```
size_t fwrite(  
const void * записываемое_данные,  
size_t размер_элемента,  
size_t число_элементов,  
FILE *указатель_на_файл );
```

```
size_t fread(void * считываемое_данное,  
size_t размер_элемента,  
size_t число_элементов,  
FILE *указатель_на_файл);
```

```
int _fileno(FILE * указатель_на_файл);
```

```
long _filelength(int дескриптор);
```

```
errno_t _chsize_s( int дескриптор,  
                  int размер );
```

```
int fseek(FILE * указатель_на_файл,  
long int число_байт, int точка_отсчета);
```

**SEEK\_SET**

**SEEK\_CUR**

**SEEK\_END**



**Пример:** Отсортировать данные в файле  
по возрастанию.

```
fopen_s(&fl, "f:\\lab1.dat", "rb+");  
int nb = sizeof(int), a, b, nwrt;  
int n = _filelength(_fileno(fl)) / nb;  
  
for (int i = 0; i < n - 1; i++)  
for (int j = i + 1; j < n; j++) {  
    fseek(fl, i*nb, SEEK_SET);  
        nwrt = fread(&a, nb, 1, fl);  
    fseek(fl, j*nb, SEEK_SET);  
        nwrt = fread(&b, nb, 1, fl);  
}
```

```
if (a>b)
{
    fseek(fl, i*nb, SEEK_SET);
    nwrt = fwrite(&b, nb, 1, fl);
    fseek(fl, j*nb, SEEK_SET);
    nwrt = fwrite(&a, nb, 1, fl);
} }
```

```
fclose(fl);
```

**Пример:** Написать программу для работы с бинарным файлом, содержащим список жильцов многоквартирного дома. Каждый элемент списка содержит следующую информацию: фамилия, номер квартиры, количество комнат в квартире. Найти и вывести на экран текстовый файл, содержащий информацию о жильцах, проживающих в трехкомнатных квартирах.

```
#include <iostream>
#include <stdio.h>
#include <io.h>
using namespace std;
```

```
FILE* fl;
```

```
struct TOwners
```

```
{
    char fio[50];
    int num;
    int nrooms;
} *list, owner;
```

```
char fname[20] = "";
```

```
int n = 0;
```

```
void fadd();           // Ввести список
```

```
void frd();           // Прочитать список
```

```
void rezc();          // Вывести результат на экран
```

```
void rezf();          // Вывести результат в файл
```

```
int menu();           // Меню
```

```
bool flopen(const char*); // Работа с файлом
```

```
int main() {  
    while (true) {  
        switch (menu()) {  
            case 1: fadd(); break;  
            case 2: frd(); break;  
            case 3: rezc(); break;  
            case 4: rezf(); break;  
            case 5: return 0;  
            default: "Error!";  
        }  
        system("pause");  
        system("cls"); } return 0;}
```

```
int menu()
{
    cout << "Select:" << endl;
    cout << "1. Enter data and write to file" << endl;
    cout << "2. Read data from file" << endl;
    cout << "3. Display the result on the screen" <<
endl;
    cout << "4. Output the result to a file" << endl;
    cout << "5. Exit" << endl;
    int i;    cin >> i;
    return i;
}
```



```
bool fopen(const char* r)
{
    if (!strlen(fname)) {
        cout << "Enter file name" << endl;
        cin >> fname;
    }

    if (fopen_s(&fl, fname, r)) {
        cout << "Error" << endl;
        return false;
    }
    else return true;
}
```

```
void fadd() {  
    if (!flopen("ab+")) return;  
    int i, n;  
    cout << "Enter the number of owners";  
    cin >> n;  
    for (i = 0; i < n; i++) {  
        cout << "Enter name: ";          cin >> owner.fio;  
        cout << "Enter apartment number:"; cin >> owner.num;  
        cout << "Enter the number of rooms:";  
            cin >> owner.nrooms;  
        fwrite(&owner, sizeof(TOwners), 1, fl);  
    }  
    fclose(fl); }
```

```
void frd()
{
    if (!flopen("rb")) return;
    n = _filelength(_fileno(fl)) / sizeof(TOwners);
    list = new TOwners[n];
    fread(list, sizeof(TOwners), n, fl);
    for (int i = 0; i < n; i++)
        cout << endl << list[i].fio << " Nomer kvartiry - " <<
            list[i].num << " Chislo komnat- " << list[i].nrooms;
    cout << endl;
    delete[]list;
    fclose(fl);
}
```

```
void rezc()
{
    if (!flopen("rb")) return;
    n = _filelength(_fileno(fl)) / sizeof(TOwners);
    for (int i = 0; i < n; i++)
    {
        fread(&owner, sizeof(TOwners), 1, fl);
        if (owner.nrooms == 3)
            cout << owner.fio << " ,apartment number - "
                << owner.num << endl;
    }
    fclose(fl);
}
```

```
void rezf()
{
    char fnamet[20];
    cout << "Enter the name of the text file" << endl;
    cin >> fnamet;
    FILE* ft;
    if (fopen_s(&ft, fnamet, "w")) {
        cout << "File not created";
        return;
    }
}
```

```
if (!flopen("rb")) return;
n = _filelength(_fileno(fl)) / sizeof(TOwners);
for (int i = 0; i < n; i++)
{
    fread(&owner, sizeof(TOwners), 1, fl);
    if (owner.nrooms == 3)
        fprintf(ft, "%s, apartment number - %d\n",
                owner.fio, owner.num);
}

fclose(fl);
fclose(ft);
}
```