

## Практическая работа №4

### Работа с файлами

Для программиста открытый файл представляется как последовательность считываемых или записываемых данных. При открытии файла с ним связывается поток ввода-вывода. Выводимая информация записывается в поток, вводимая информация считывается из потока.

Когда поток открывается для ввода-вывода, он связывается со стандартной структурой типа FILE, которая определена в `stdio.h`. Структура FILE содержит необходимую информацию о файле.

Открытие файла осуществляется с помощью функции `fopen()`, которая возвращает указатель на структуру типа FILE, который можно использовать для последующих операций с файлом.

```
FILE *fopen(name, type);
```

`name` — имя открываемого файла (включая путь),

`type` — указатель на строку символов, определяющих способ доступа к файлу:

"r" — открыть файл для чтения (файл должен существовать);

"w" — открыть пустой файл для записи; если файл существует, то его содержимое теряется;

"a" — открыть файл для записи в конец (для добавления); файл создается, если он не существует;

"r+" — открыть файл для чтения и записи (файл должен существовать);

"w+" — открыть пустой файл для чтения и записи; если файл существует, то его содержимое теряется;

"a+" — открыть файл для чтения и дополнения, если файл не существует, то он создаётся.

Возвращаемое значение — указатель на открытый поток. Если обнаружена ошибка, то возвращается значение NULL.

Функция `fclose()` закрывает поток или потоки, связанные с открытыми при помощи функции `fopen()` файлами. Закрываемый поток определяется аргументом функции `fclose()`.

Возвращаемое значение: значение 0, если поток успешно закрыт; константа EOF, если произошла ошибка.

Чтение символа из файла:

```
char fgetc(поток);
```

Аргументом функции является указатель на поток типа FILE. Функция возвращает код считанного символа. Если достигнут конец файла или возникла ошибка, возвращается константа EOF.

Запись символа в файл:

```
fputc(символ, поток);
```

Аргументами функции являются символ и указатель на поток типа FILE. Функция возвращает код считанного символа.

Функции `fscanf()` и `fprintf()` аналогичны функциям `scanf()` и `printf()`, но работают с файлами данных, и имеют первый аргумент — указатель на файл.

```
fscanf(поток, "ФорматВвода", аргументы);  
fprintf(поток, "ФорматВывода", аргументы);
```

Функции `fgets()` и `fputs()` предназначены для ввода-вывода строк, они являются аналогами функций `gets()` и `puts()` для работы с файлами.

```
fgets(УказательНаСтроку, КоличествоСимволов, поток);
```

Символы читаются из потока до тех пор, пока не будет прочитан символ новой строки `'\n'`, который включается в строку, или пока не наступит конец потока EOF или не будет прочитано максимальное символов. Результат помещается в указатель на строку и заканчивается нуль-символом `'\0'`. Функция возвращает адрес строки.

```
fputs(УказательНаСтроку, поток);
```

Копирует строку в поток с текущей позиции. Завершающий нуль-символ не копируется.

#### **Пример:**

Ввести число и сохранить его в файле `s1.txt`. Считать число из файла `s1.txt`, увеличить его на 3 и сохранить в файле `s2.txt`.

```
#include <stdio.h>  
int main()  
{  
    FILE *S1, *S2;  
    int x, y;  
    printf("Введите число : ");  
    scanf("%d", &x);  
    S1 = fopen("S1.txt", "w");  
    fprintf(S1, "%d", x);  
    fclose(S1);  
    S1 = fopen("S1.txt", "r");  
    S2 = fopen("S2.txt", "w");  
    fscanf(S1, "%d", &y);  
    y += 3;  
    fclose(S1);  
    fprintf(S2, "%d\n", y);  
    fclose(S2);  
}
```

#### **Задание на практику:**

- получить с клавиатуры число N;
- создать массив размером N, заполненный случайными числами;
- записать этот массив в файл.