

# Лекция 6

Алгоритмические языки и  
программирование

# Работа с файлами в языке Си

- **Файл** – именованная область внешней памяти, выделенная для хранения массива данных. Данные, содержащиеся в файлах, имеют самый разнообразный характер: программы на алгоритмическом или машинном языке; исходные данные для работы программ или результаты выполнения программ; произвольные тексты; графические изображения и т. п.

# Работа с файлами в языке Си

- 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.

# Функция fopen()

- Открытие файла осуществляется с помощью функции fopen(), которая возвращает указатель на структуру типа FILE, который можно использовать для последующих операций с файлом.
- Возвращаемое значение - указатель на открытый поток. Если обнаружена ошибка, то возвращается значение NULL.

# Функция fopen()

- FILE \*fopen(name, type);
- name – имя открываемого файла (включая путь),
- type - указатель на строку символов, определяющая 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.

# Функция `fclose()`

- 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.

# Функция `getc()`

- `int getc(FILE *fp)`
- Функция `getc` возвращает следующий СИМВОЛ ИЗ ПОТОКА, на который указывает `*fp`; в случае исчерпания файла или ошибки она возвращает EOF.

# Функция puts()

Функция puts пишет символ с в файл fp

```
int puts(int c, FILE *fp)
```

и возвращает записанный символ или EOF в случае ошибки.



# Ввод и вывод файлов

- Форматный ввод-вывод файлов можно построить на функциях `fscanf` и `fprintf`. Они идентичны `scanf` и `printf` с той лишь разницей, что первым их аргументом является указатель на файл, для которого осуществляется ввод-вывод, формат же указывается вторым аргументом.
- `int fscanf(FILE *fp, char *format, ...)`
- `int fprintf(FILE *fp, char *format, ...)`

# feof()

Функция `feof(FILE *)` возвращает ненулевое значение, если встретился конец указанного в аргументе файла.

```
int feof(FILE *fp)
```

# Пример

```
#include <stdio.h>
#include <locale.h>
int main () {
FILE *fp, *fp2;
int sym;
fp = fopen("test.txt", "r");
fp2 =fopen("test2.txt", "w+");
fscanf(fp, "%c", &sym);
if ((fp !=NULL) && (fp2!=NULL)) { // открыть файлы удалось
    while (feof(fp)==0){
        fprintf (fp2, "Я прочел %c \n", sym);
        fscanf(fp, "%c", &sym);
    }
} else printf("Не удалось открыть файл");
fclose(fp);
fclose(fp2);
return 0;
}
```

# Файлы

- Напишите программу, которая считывает слова из одного файла и перемещает каждое слово с новой строки и с нумерацией.

