

ЛАБОРАТОРНАЯ РАБОТА № 11

Работа с текстовыми файлами.

1. Краткие теоретические сведения

1.1. Организация работы с файлами

Различают два вида файлов: текстовые и двоичные.

Текстовые файлы хранят информацию в виде последовательности символов. В текстовом режиме каждый разделительный символ строки автоматически преобразуется в пару (возврат каретки – переход на новую строку).

Бинарные (или двоичные) файлы предназначены для хранения только числовых значений данных. Структура такого файла определяется программно.

Функции для работы с файлами размещены в библиотеках `stdio.lib` (`#include <stdio.h>`). Каждый файл должен быть связан с некоторым указателем. Этот указатель имеет тип `FILE` и используется во всех операциях с файлами.

Формат объявления указателя на файл следующий:

FILE *указатель на файл;

Макрос **NULL** определяет пустой указатель.

Макрос **EOF**, часто определяемый как `-1`, является значением, возвращаемым тогда, когда функция ввода пытается выполнить чтение после конца файла.

1.2. Функции для работы с файлами

Функция `FILE *fopen(const char *имя_файла, const char *режим_открытия);`

открывает файл и связывает его с потоком. Возвращает указатель на открытый файл. *Имя_файла* – это указатель на строку символов, в которой хранится имя файла и путь к нему. *Режим_открытия* – это указатель на строку символов, в которой указывается режим открытия файла. Допустимые режимы:

r – открытие текстового файла для чтения;

w – создание текстового файла для записи;

a – добавление информации в конец текстового файла.

При работе с текстовыми файлами к символу, указывающему режим открытия, добавляется символ «*t*» (по умолчанию), а при работе с бинарными – «*b*». Если необходимо и читать и записывать в файл, то добавляется символ «*+*». При возникновении ошибки во время открытия файла, функция **fopen** возвращает значение **NULL**.

Функция `int fclose(FILE *указатель_на_файл);`

закрывает поток, который был открыт с помощью вызова **fopen()** и записывает в файл все данные, которые еще оставались в дисковом буфере. Доступ к файлу после выполнения функции будет запрещен.

Возвращение нуля означает успешную операцию закрытия. В случае же ошибки возвращается **EOF**.

Функция `int fcloseall(void);`

закрывает все открытые файлы. Возвращает количество закрытых файлов или **EOF**, если возникает ошибка.

Функция `int fputc(int символ, FILE * указатель_на_файл);`

записывает один символ в текущую позицию указанного открытого файла.

Функция `int fgetc(FILE * указатель_на_файл);`

читает один символ из текущей позиции указанного открытого файла.

Функция `int feof(FILE * указатель_на_файл);`

возвращает отличное от нуля значение (`true`), если конец файла не достигнут, и ноль (`false`), если достигнут конец файла.

Для построчного ввода-вывода :

Функция **int fputs(const char * строка, FILE * указатель_на_файл);**
записывает строку символов в текущую позицию указанного открытого файла.
Прием символов заканчивается после передачи n байтов или при получении "\n".
Управляющий символ "\n" тоже передается в принимающую строку. В любом случае строка заканчивается "\0". При успешном завершении считывания, функция возвращает указатель на прочитанную строку, иначе возвращает NULL.

Функция **char *fgets(char * строка, int длина, FILE * указатель_на_файл);**
читает строку символов из текущей позиции указанного открытого файла до тех пор, пока не будет прочитан символ перехода на новую строку, или количество прочитанных символов не станет равным *длина* - 1.

Функция **int *fprintf(FILE * указатель_на_файл, const char * управляющая_строка);**
записывает форматированные данные в файл. *Управляющая_строка* определяет строку форматирования аргументов, заданных своими адресами. Обычно эта строка состоит из последовательности символов «%», после которых следует символ типа данных:

Функция **int *fscanf(FILE * указатель_на_файл, const char * управляющая_строка);**
читает форматированные данные из файла.

Строка форматирования строится аналогично функции **fprintf. (printf)**

Функция **void rewind(FILE * указатель_на_файл);**
устанавливает указатель текущей позиции выделенного файла в начало файла.

Функция **int ferror(FILE * указатель_на_файл);**
определяет, произошла ли ошибка во время работы с файлом.

Функция

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

записывает в файл заданное число данных определенного размера. Размер данных задается в байтах. Тип **size_t** определяется как целое без знака.

Функция

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

считывает из файла указанное число данных заданного размера. Размер задается в байтах. Функция возвращает число прочитанных элементов. Если число прочитанных элементов не равно заданному, то при чтении возникла ошибка или встретился конец файла.

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

устанавливает указатель в заданную позицию. Заданное количество байт отсчитывается от позиции, которая задается следующими макросами: **SEEK_SET** - начало файла, **SEEK_CUR** - текущая позиция, **SEEK_END** - конец файла.

1.3. Пример (фрагмент кода)

```
int MAXLINE=255; //максимальная длина строки
FILE *in, //исходный файл
*out; //принимающий файл
char buf[MAXLINE]; //строка, с помощью которой выполняется копирование
//копирование строк одного файла в другой
while (fgets (buf, MAXLINE, in)!=NULL)
fputs(buf,out);
```

2. Индивидуальные задания

2.1 ПОСТАНОВКА ЗАДАЧИ

1. Создать текстовый файл F1 не менее, чем из 7 строк и записать в него информацию
2. Выполнить задание по вариантам.

2.2. ВАРИАНТЫ .

1. Скопировать в файл F2 только четные строки из F1.
Подсчитать размер файлов F1 и F2 (в байтах).
2. Скопировать в файл F2 только те строки из F1, которые начинаются с буквы «А».
Подсчитать количество слов в F2.
3. Скопировать в файл F2 только те строки из F1, которые начинаются и заканчиваются на одну и ту же букву.
Подсчитать количество символов в F2.
4. Скопировать из файла F1 в файл F2 строки, начиная с 4.
Подсчитать количество символов в последнем слове F2.
5. Скопировать из файла F1 в файл F2 строки, начиная с K до K+5.
Подсчитать количество гласных букв в файле F2.
6. Скопировать из файла F1 в файл F2 строки, начиная с N до K.
Подсчитать количество согласных букв в файле F2.
7. Скопировать из файла F1 в файл F2 все строки, кроме тех, что начинаются на букву А.
Подсчитать количество символов в первом слове F2.
8. Скопировать из файла F1 в файл F2 все строки, которые не содержат цифры.
Подсчитать количество строк, которые начинаются на букву «А» в файле F2.
- 9 Скопировать из файла F1 в файл F2 все строки, которые содержат только одно слово.
Найти самое длинное слово в файле F2.
10. Скопировать из файла F1 в файл F2 все строки, которые не содержат слова ,
начинающиеся на одну букву.
Найти самое короткое слово в файле F2.
11. Скопировать из файла F1 в файл F2 все строки, кроме той строки, которая содержит
самое короткое слово.
Напечатать номер этой строки.
12. Скопировать из файла F1 в файл F2 все строки, кроме той строки, в которой больше
всего гласных букв.
Напечатать номер этой строки.
13. Скопировать из файла F1 в файл F2 все строки, начинающиеся на букву «А»
и расположенные между строками с номерами N1 и N2.
Определить номер той строки, в которой больше всего согласных букв, файла F2.

14. Скопировать из файла F1 в файл F2 все строки, не содержащие букву «А» и расположенные между строками с номерами N1 и N2.
Определить номер той строки, в которой больше всего гласных букв, файла F2.
15. Скопировать из файла F1 в файл F2 все строки, заканчивающиеся на букву «А» и расположенные между строками с номерами N1 и N2.
Определить номер той строки, в которой больше всего букв «А», файла F2.
16. Скопировать из файла F1 в файл F2 все строки, начинающиеся на букву «А» и заканчивающиеся на букву «С»,
расположенные между строками с номерами N1 и N2.
Определить количество слов в первой строке файла F2.
17. Скопировать из файла F1 в файл F2 все строки, начинающиеся на букву «А» расположенные между строками с номерами N1 и N2, а затем все строки от N2+3 и до последней.
Определить количество слов в последней строке файла F2.
18. Скопировать из файла F1 в файл F2 все строки, в которых нет одинаковых слов.
Определить количество гласных букв в первой строке файла F2.
19. Скопировать из файла F1 в файл F2 все строки, в которых нет слов, совпадающих с первым словом.
Определить количество согласных букв в первой строке файла F2.
20. Скопировать из файла F1 в файл F2 все строки, в которых есть одинаковые слова.
Определить количество гласных букв в последней строке файла F2.
21. Скопировать из файла F1 в файл F2 все строки, в которых есть слова, совпадающие с первым словом.
Определить количество согласных букв в последней строке файла F2.
22. Скопировать из файла F1 в файл F2 все строки, в которых более 2 слов.
Определить номер слова, в котором больше всего гласных букв.
23. Скопировать из файла F1 в файл F2 все строки, в которых содержится только одно слово.
Определить номер слова, в котором больше всего согласных букв.
24. Скопировать из файла F1 в файл F2 все строки, в которых содержится два одинаковых слова.
Определить номер слова, в котором больше всего букв «А».
25. Скопировать из файла F1 в файл F2 все строки, в которых содержится не менее двух одинаковых слов.
Определить номер слова, в котором больше всего цифр.