**Министерство науки и высшего образования РФ  
ФГБОУ ВО  
Московский авиационный институт**

**(национальный исследовательский университет)**

Институт №3.

Системы управления, информатика и электроэнергетика.

Кафедра 304.

**Отчет по лабораторной работе №9**

**по учебной дисциплине**

**«Программирование на языке высокого уровня»**

**на тему**

***«Изучение принципов работы с двоичными файлами»***

Группа: *М3О-209Б-19*

*Вариант №4*

Выполнил: студент группы М3О-209б-19 *Кузнецов И.И.*

Приняли: **Доцент к. 304, к.т.н., Новиков П. В.**

**Старший преподаватель к. 304, Ивашенцев И. В.**

Москва 2020

**Министерство науки и высшего образования РФ  
ФГБОУ ВО  
Московский авиационный институт**

**(национальный исследовательский университет)**

Институт №3.

Системы управления, информатика и электроэнергетика.

Кафедра 304.

**Отчет по лабораторной работе №9**

**по учебной дисциплине**

**«Программирование на языке высокого уровня»**

**на тему**

***«Изучение принципов работы с двоичными файлами»***

Группа: *М3О-209Б-19*

*Вариант №4*

Выполнил: студент группы М3О-209б-19 *Мухаметгалиев А.Р.*

Приняли: **Доцент к. 304, к.т.н., Новиков П. В.**

**Старший преподаватель к. 304, Ивашенцев И. В.**

Москва 2020

**Содержание**

1. **Задание.**
2. **Структурная схема алгоритма программы и подпрограмм.**
3. **Текст программы.**
4. **Результаты работы программы.**
5. **Вывод.**

**Задание**

**Вариант 4**

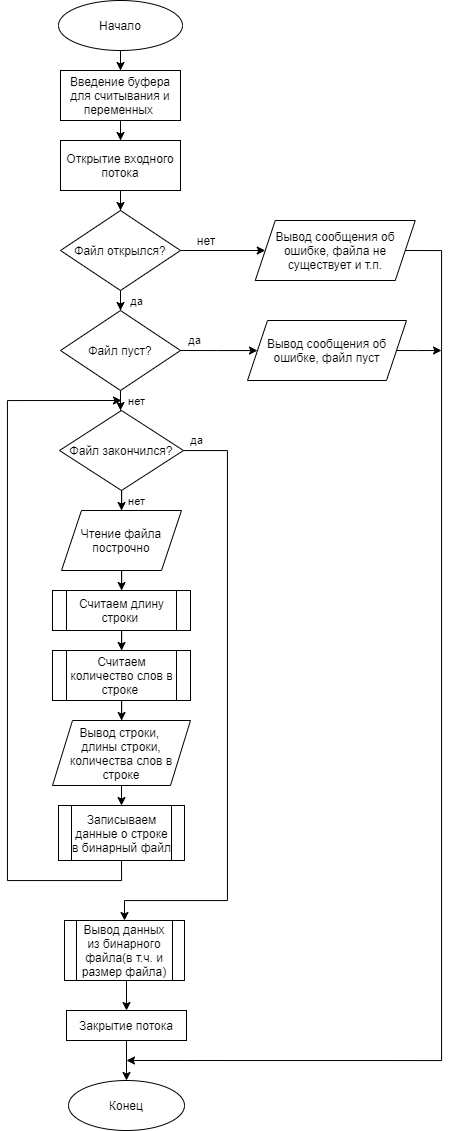
1. Разработать программу и подпрограмму (подпрограммы), работающую с текстовым файлом и выполняющую действия согласно варианту задания:

Преобразовать входной текстовый файл в выходной двоичный, содержащий записи следующего вида:

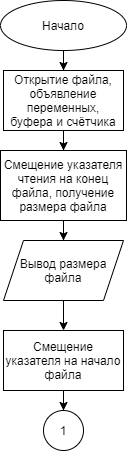
* + данные типа float – длина строки;
  + данные типа int – количество слов в строке;
  + массив типа char – строка входного файла без завершающего символа '\n'.

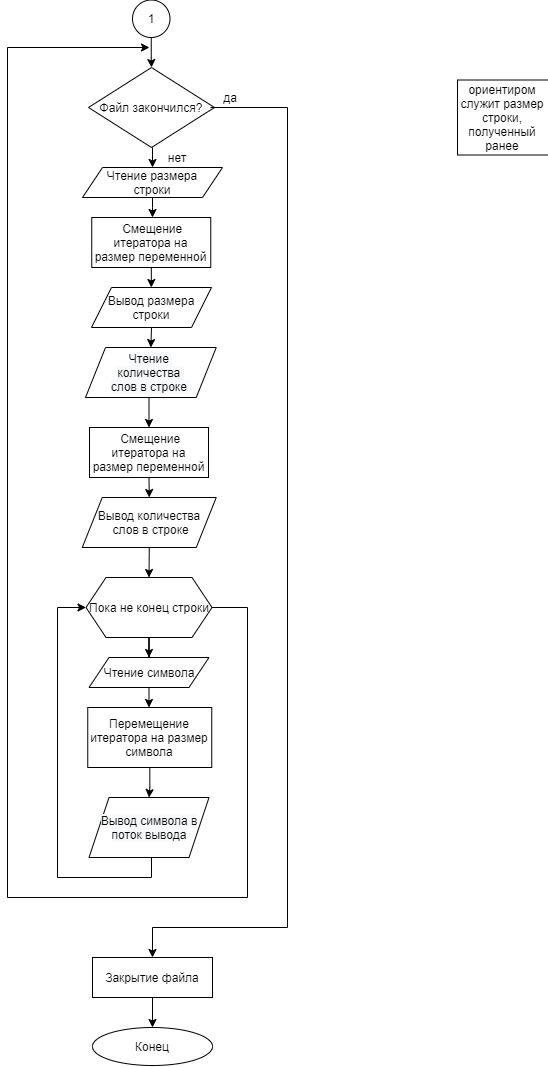
1. Разработать программу, которая выводит на экран размер и содержимое двоичного файла заданной структуры, созданного в соответствии с заданием из п.1.

**Структурная схема алгоритма программы.**

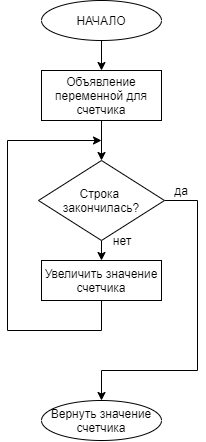


**Структурная схема алгоритма подпрограммы readFromFile**

****

****

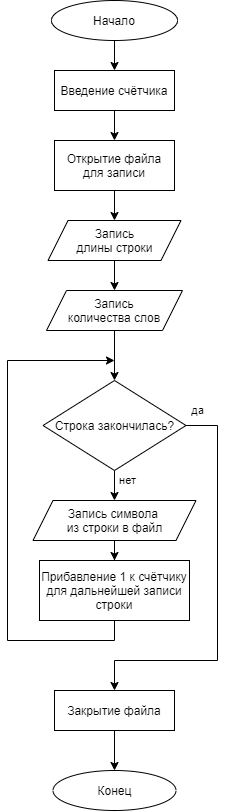
**Структурная схема алгоритма подпрограммы countLenght.**

****

**Структурная схема алгоритма подпрограммы wordCounter.**

****

**Структурная схема алгоритма подпрограммы binarySingle.**

****

**Текст программы**

#include <stdio.h>

int wordCounter(const char\* str);

float countLength(const char\* str);

void binarySingle(const char\* str, float lengthString, int numberWords);

void readFromFile();

int main() {

FILE\* in;

in = fopen("../Fin.txt", "rt");

char str[256]; //буфферная строка

int numberWords; //количество слов в строке

float lengthString; //длина строки

if (in == NULL) { //проверка на наличие файла

perror("ERROR with a reading file ");

return -1;

}

if (getc(in) == EOF) { //проверка на пустоту файла

printf("File is empty!\n");

return -2;

}

rewind(in); //переход на начало файла

remove("../Fout.bin");

while (fgets(str, sizeof(str), in) != NULL) { //считать построчно, пока

//не конец файла

lengthString = countLength(str); //считаем длину строки

numberWords = wordCounter(str); //считаем количество слов в строке

printf("The string is: \n%s", str); //вывод строки

printf("The string's size is: %f\n", lengthString); //вывод длины

//строки

printf("The number of words: %d\n\n", numberWords); //вывод

//количества слов в строке

binarySingle(str, lengthString, numberWords); //перевод строки в

//бинарный файл

}

printf("------------------------------------\n");

readFromFile(); //считаем размер файла в

//байтах и выводим всю информацию из него

fclose(in);

return 0;

}

int wordCounter(const char\* str) {

int counter = 0;

int i = 0;

while (str[i] != '\0') { //проходимся по всей строке

while (str[i] == ' ') i++; //пропускаем пробелы

if (str[i + 1] == '\0') return 0; //если кроме пробелом ничего нет

while (str[i] != ' ' && str[i] != '\0') i++; //находим слово идем по

//нему и

counter++; //добавляем к общему количеству

while (str[i] == ' ') i++; //пропускаем пробелы

}

return counter;

}

float countLength(const char\* str) { //подсчет длины строки

int i = 0;

while (str[i] != '\0' && str[i] != '\n') i++;

return i;

}

void binarySingle(const char\* str, float lengthString, int numberWords) {

int i = 0;

FILE\* out;

out = fopen("../Fout.bin", "a+b");

fwrite(&lengthString, sizeof(float), 1, out); //записываем длину строку

fwrite(&numberWords, sizeof(int), 1, out); //записываем количество слов

while (str[i] != '\0' && str[i] != '\n') { //записываем строку

fwrite(&str[i], sizeof(char), 1, out);

i++;

}

fclose(out);

}

void readFromFile() {

FILE\* rebin;

rebin = fopen("../Fout.bin", "rb");

int sizeOfFile; //размер файла

float lengthString; //длина строки

int numberWords; //количество слов в строке

char symbol; //буферный символ

int i = 0; //итератор

fseek(rebin, 0, SEEK\_END); //смещение указателя чтения на конец файла

sizeOfFile = ftell(rebin); //получение размера файла

printf("The size of a file is: %u bytes \n\n", sizeOfFile);

fseek(rebin, 0, SEEK\_SET); //смещение указателя на начало файла

while (i < sizeOfFile) { //читаем до конца файла

fread(&lengthString, sizeof(int), 1, rebin); //чтение размера строки

//из файла

i += sizeof(float); //смещение итератора на размер переменной

printf("The length of this string is: %f \n", lengthString);

fread(&numberWords, sizeof(int), 1, rebin); //чтение количества слов

//из файла

i += sizeof(int); //смещение итератора на размер переменной

printf("The number of words is: %u \n", numberWords);

printf("%s", "The string is: ");

for (int j = 0; j < lengthString; ++j) {

fread(&symbol, sizeof(char), 1, rebin);

i += sizeof(char);;

putchar(symbol);

}

printf("\n\n");

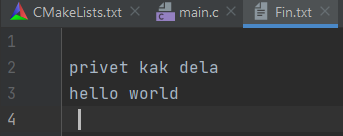
}

fclose(rebin);

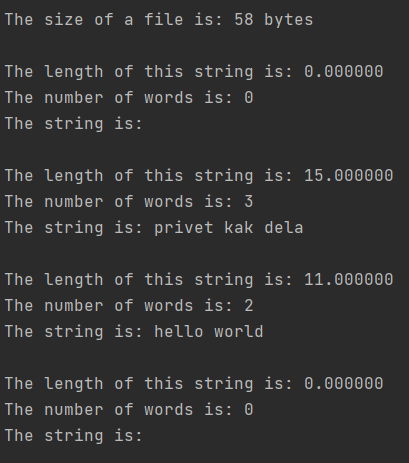
}

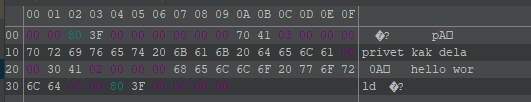
**Результаты работы программы**

Входные данные:



Результат:





**Вывод**

Мы разработали программу и подпрограммы:

1. работающие с текстовым файлом и выполняющую действия согласно варианту задания:

Преобразовать входной текстовый файл в выходной двоичный, содержащий записи следующего вида:

* + данные типа float – длина строки;
  + данные типа int – количество слов в строке;
  + массив типа char – строка входного файла без завершающего символа '\n'.

1. Которая выводит на экран размер и содержимое двоичного файла заданной структуры, созданного в соответствии с заданием из п.1.